AD-A278 026

(2)

RL-TR-93-258, Vol II (of two)
Final Technical Report
December 1993

INVESTIGATION AND SIMULATION OF NONLINEAR PROCESSORS FOR SPREAD SPECTRUM RECEIVERS, USERS MANUAL

Illinois Institute of Technology

Donald R. Ucci, William Jacklin, and Jimm Grimm

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

Although this report references * limited document listed on page 18, no limited information has been extracted.

RL-TR-93-258, Vol II (of two) has been reviewed and is approved for publication.

APPROVED:

JOHN J. PATTI
Project Engineer

FOR THE COMMANDER

JOHN A. GRANIERO
Chief Scientist for C3

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1993	Final Feb 92 - Aug 93

4. TITLE AND SUBTITLE

INVESTIGATION AND SIMULATION OF NONLINEAR PROCESSORS FOR SPREAD SPECTRUM RECEIVERS, USERS MANUAL

5. FUNDING NUMBERS

C - F30602-92-C-0039
PE - 62702F
PR - 4519
TA - 42
WU - PA

6. AUTHOR(S)

Donald R. Ucci, William Jacklin, and Jimm Grimm

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Illinois Institute of Technology
Electrical and Computer Engineering Department
Chicago IL 60616-3793

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Rome Laboratory (C3BB)
525 Brooks Road
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-93-258, Vol II (of two)

11. SUPPLEMENTARY NOTES

Rome Laboratory Project Engineer: John J. Patti/C3BB/(315) 330-3615

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The objective of the recent research effort was to investigate and determine the viability of utilizing Locally Optimal (LO) nonlinear processing to mitigate non-Gaussian interfering signals in a Direct Sequence (DS) SS communications system. The effort centered on the use of memoryless techniques, as well as techniques employing memory, and performance comparisons of many receiver and nonlinear processor configurations. The approach used included the analysis and evaluation of several implementation of the various nonlinear processing algorithms. The analysis included the study of well known techniques, as well as newly developed methods. Evaluation was accomplished through the development of software simulations designed to test the algorithms in various signalling scenarios. The results illustrate the tradeoffs of each nonlinear processor algorithm for use in a spread spectrum receiver. This knowledge can be used to determine the most effective processor for a given interference scenario. The work presented in this report is directly in line with the mission of Rome Laboratory (RL) to provide secure, reliable communications to the United States Air Force.

14. SUBJECT TERMS

Adaptive Filtering, Nonlinear Processing, Spread Spectrum

15. NUMBER OF PAGES

152

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

Volume II

1. Introduction

This volume of the report serves as a user's guide to the Illinois Institute of Technology (IIT) Signal Processing Worksystem™ (SPW) simulations. These simulations consist of nine modular SPW systems and fifteen Memoryless Nonlinear Processor (MNP) blocks for use with these systems. ***A solid understanding of the concepts discussed in Volume I of this report is essential for using these systems and interpreting the results.***

NOTICE TO THE USER:

The most important system parameter relationships in this report are summarized in three text boxes like this one. It is imperative that these fundamental constraints are maintained at all times during simulation of the systems. There are no provisions in the systems to automatically verify these constraints; it is entirely the responsibility of the user. If these constraints are violated, invalid results may be produced.

2. The IIT SPW MNPs

In the current and past research efforts five types of MNPs have been implemented in the SPW platform: Histogram [Illi93a], Equiprobable Bin Histogram (EBH) [Illi93b], Fourier Series Approximation (FSA) [Illi93b], Continuous Polynomial Approximation (CPA) [Grim93], and M Interval Polynomial Approximation (MIPA) [Illi91]. Table 1 summarizes the filenames

for the MNPs[1] and the linear receiver. These filenames are for reference only since they may be copied directly from the SPW User Palette [Comd91] into the desired system.

Filenames of MNP Blocks

NONLINEAR RECEIVERS WITH CORRELATOR:

Library/Filename	Description
HISTO/SERIAL_NONLIN	Histogram MNP
HISTO/S_ENONLIN	EBH MNP
FOURIER/SERIAL_NONLIN	FSA MNP
CPA/S_CDF2_NONLIN	CPA with Linear Transform MNP
CPA/S_CDF3_NONLIN	CPA with Discontinuous Auxiliary Function MNP
CPA/S_PDF_GT_NONLI	CPA with Gaussian Tails MNP
MIPA/SERIAL_NONLIN	MIPA MNP (for 2nd and 4th order MIPA)
LINEAR/SERIAL_LIN	Linear receiver

NONLINEAR RECEIVERS WITHOUT CORRELATOR:

Library/Filename	Description
HISTO/S_NONLIN_NC	Histogram MNP
HISTO/S_ENONLIN_NC	EBH MNP
FOURIER/S_NONLIN_NC	FSA MNP
CPA/S_CDF2_NL_NC	CPA with Linear Transform MNP
CPA/S_CDF3_NL_NC	CPA with Discontinuous Auxiliary Function MNP
CPA/S_PDF_GT_NL_NC	CPA with Gaussian Tails MNP
MIPA/S_NONLIN_NC	MIPA MNP (for 2nd and 4th order MIPA)

Table 1

[1] In this report SPW filenames are written as LIBRARY/FILENAME, where LIBRARY is the Block Diagram Editor (BDE) library in which the block or system is stored, and FILENAME is the name of the block or system.

There are two main categories of MNPs: MNPs with a correlator and MNPs without a correlator. The MNPs with a correlator implement the entire memoryless Locally Optimal (LO) algorithm [Illi93b]:

Choose $(\overline{s}_{I_m}, \overline{s}_{Q_m})$ which maximizes:

$$l_m = \sum_{k=1}^{N} \left\{ s_{I_m} g(r_k) \cos\theta_k + s_{Q_m} g(r_k) \sin\theta_k \right\} \tag{1}$$

where $g(r_k) = -\dfrac{\frac{d}{dr_k} f_R(r_k)}{f_R(r_k)} + \dfrac{1}{r_k}$ is the LO Memoryless Nonlinear Transform (MNT).

Since the correlator performs a decision, these blocks constitute the entire nonlinear receiver, and may be used in systems such as the conventional Quadrature Phase Shift Keying (QPSK) [Taub86] systems to be discussed in Section 3. It is important to make the distinction between the MNT, which is the mathematical formula, and the MNP, which is the block that applies the MNT.

The MNPs without a correlator only implement the MNT and convert it to rectangular coordinates as follows:

$$output = [g_{I_k} \;\; g_{Q_k}]^T = [s_{I_m} g(r_k) \cos\theta_k \quad s_{Q_m} g(r_k) \sin\theta_k]^T \tag{2}$$

These blocks may be used in systems where additional post-processing is required after the MNP, as in the Spread Spectrum (SS) systems to be discussed in Section 3.

3. The IIT SPW Systems

Nine modular SPW systems have been designed to incorporate the MNPs previously discussed. Four of these systems are conventional QPSK systems and five are Direct Sequence (DS) SS [Taub86] QPSK systems. The conventional systems aid in isolating and characterizing the performance of the MNPs, and the DSSS systems help analyze the performance of the MNPs in a spread spectrum environment. Both the conventional and the DSSS systems are based on QPSK modulation; however, it would be straightforward to modify them to employ alternative modulation schemes such as Minimum Shift Keying, Quadrature Amplitude Modulation, or M-ary Phase Shift Keying. The filenames and purposes of the IIT SPW systems are summarized in Table 2. All systems simulate a communication system with jammers and nonlinear processing, and each one has certain processing after the nonlinearity that is also listed in the table.

Figure (1) shows the basic conventional QPSK system, SIM/PE, which calculates the probability of bit error (P_b) for any of the serial MNPs or the linear receiver. Since QPSK is a basic form of modulation it was chosen to isolate the effect of the MNP on the overall system performance. The RL/COMPLEX_DATA block generates random QPSK data, and the JAM/CHANNEL block adds the jammers and Gaussian noise. An MNP or the linear receiver block must be copied by the user to the designated space in Fig. (1). The HISTO/PROB_OF_ERROR block counts the number of errors the receiver makes in decoding the bits and computes P_b.

The SIM/PES system shown in Fig. (2) is able to calculate P_b for up to three different receiver configurations simultaneously to facilitate comparisons of the MNPs. The SIM/PE and SIM/PES simulations may be used in conjunction with the SPW iteration macro [Varn93] to generate Bit Error Rate (BER) curves as a function of various system parameters.

Filenames of IIT SPW Systems

CONVENTIONAL QPSK SIMULATIONS:

Library/Filename	Description
SIM/PE	Computes P_b for any MNP
SIM/PES	Computes P_b for any three MNPs
SIM/HSIGS	Computes P_b & plots signals for histogram MNPs
SIM/PSIGS	Computes P_b & plots signals for polynomial and FSA MNPs

QPSK SPREAD SPECTRUM SIMULATIONS:

Library/Filename	Description
SIM/SS_PE	Computes P_b for any MNP
SIM/SS_PES	Computes P_b for any three MNPs
SIM/DSSSH	Computes P_b & plots signals for histogram MNPs
SIM/DSSSP	Computes P_b & plots signals for polynomial and FSA MNPs
SIM/DSSSL	Computes P_b & plots signals for linear system

Table 2

The other two conventional systems are SIM/HSIGS and SIM/PSIGS, and are displayed in Figs. (3) and (4), respectively. HSIGS is an abbreviation of "histogram signals". In addition to computing P_b, this system outputs all relevant signals for detailed analysis. The SIM/HSIGS system works for both the histogram and EBH MNPs. PSIGS is short for "polynomial signals", and the SIM/PSIGS system works for the CPA and MIPA MNPs. SIM/PSIGS also works for the FSA MNP, even though the FSA is not a polynomial approximation method.

The DSSS systems are shown in Figs. (5) through (9). The basic system, SIM/SS_PE, is shown in Fig. (5). QPSK data is generated by the SPW QPSK SOURCE library block. The spectral spreading is achieved by multiplying the QPSK data by a Pseudo Noise (PN) sequence, and the jammers and Gaussian noise are added in the channel. The user must copy an MNP without a correlator into the designated space. The transformed sequence is multiplied by the

same PN sequence to despread the spectrum. A PSK matched filter demodulator library block is used to receive the message, and the HISTO/PROB_OF_ERROR block is used to compute P_b.

4. Using the MNPs in an IIT SPW System

A simple three step process is required to use the MNPs in the IIT SPW simulations. First, the user enters the Block Diagram Editor (BDE) and loads one of the systems listed in Table 2 into a viewport. Second, the desired MNP is copied from the User Palette into the designated space in the system, making sure that all wires are properly connected. ***The parameters in the MNP block are already exported to the parameters on the top level of the system.*** Finally, the system level parameters are adjusted by the user to their desired settings and the system is simulated.

5. Parameters for IIT SPW Simulations

There are numerous parameters which specify the configuration of the IIT SPW simulations. It is not possible to specify every parameter explicitly since some parameters are dependent on others. For example, the symbol rate, R_s, the sampling frequency, f_s, and the number of samples per symbol, N_s, are related by $R_s = f_s/N_s$. Thus, specifying any two of these parameters implies a value for the third. ***In this manual a parameter which is directly specified is enclosed in braces {} for clarity. The remaining parameters are considered to be variables, and their values are a function of the specified parameters.*** A listing of the parameters of the IIT SPW simulations is presented in Table 3.

IIT SPW System Parameters

SOURCE PARAMETERS:	Symbol	Name in SPW
Probability of Zero Symbol	P_z	prob_zero
Sampling Frequency	f_s	s_freq
Samples Per Symbol	N_s	samples_per_symbol
Symbol Rate	R_s	Rs
ADDITIONAL SOURCE PARAMETERS FOR SPREAD SPECTRUM SYSTEMS:		
Samples per Chip	N_c	samples_per_chip
Processing Gain	PG	pn_gain
CHANNEL PARAMETERS:		
Continuous Wave Jammer		
Jammer to Signal Ratio	J/S	J1_S, J2_S, J3_S
Frequency Fraction	f_J/R_s	freq1, freq2, freq3
Jammer Phase	ϕ_J	phase1, phase2, phase3
Partial Band Jammer		
Jammer to Signal Ratio	J_{PB}/S	Jpb_S
Cutoff Frequency Fraction	f_c/R_s	pfreq
Gaussian Noise		
Bit Energy to Gaussian Noise Power Ratio	E_b/N_0	Eb_No
Gaussian Noise Power	N_0	No
RECEIVER PARAMETERS:		
Samples per Correlation	N	samples
Order of Approximation	P	order
Number of Bins	B	bins
Symbols per P_b calculation	N_{Pb}	symbols_per_calc

Table 3

5.1 Parameter Overview

Several of the parameters in the SPW simulations are interrelated, and certain relationships between them must be maintained. In addition, the number of iterations per simulation run is also dependent on certain parameters. It is necessary to choose the correct number of iterations per simulation to be certain that the desired number of P_b values will be calculated. ***Do not select run to EOF***, because there is no end of file in these systems and they would run indefinitely.

The nonlinearity in the MNP blocks requires vector operation. In all of the MNPs listed in Table 1 the incoming serial data is buffered, operated on by the nonlinearity in vector form, and converted back to serial form. The length of this vector is N, the number of samples per correlation. This length also corresponds to the number of samples for each MNT approximation. The correlation must be based on an integer number of symbols, so N/N_s must be an integer. If N/N_s is not an integer correlations after the first one will not be synchronized to the symbol period and P_b will suffer as a result.

Let I_Q be the number iterations for a conventional QPSK simulation run and k be the number of P_b points to be computed. Also, let C be the number of correlations per P_b calculation. This also corresponds to the number of MNT approximations per P_b calculation. C is an implicit variable which is a function of other parameters. ***The following three constraints***, **(A)**, **(B)**, ***and*** **(C)**, ***must be maintained at all times***:

$$N / N_s * C = N_{Pb} \qquad \text{(A)}$$
$$I_Q = k * N * C + N \qquad \text{(B)}$$
$$N / N_s \text{ must be an integer} \qquad \text{(C)}$$

Fundamental Constraint Box (1)

For example, let $\{N\} = 5{,}000$, $\{N_s\} = 25$ and $\{N_{Pb}\} = 1{,}000$. (These are the default values in the IIT SPW systems.) From constraint (A),

$$\{N\} / \{N_s\} * C = \{N_{Pb}\}$$
$$5{,}000 / 25 * C = 1{,}000$$
$$C = 5$$

The number of iterations required for a single P_b data point ($k = 1$) is determined from constraint (B) to be $I_Q = 25{,}000 + 5{,}000 = 30{,}000$. To compute three P_b data points requires $I_Q = 3 * 25{,}000 + 5{,}000 = 80{,}000$. Note that for both cases $C = 5$ and $\{N\} / \{N_s\} = 200$, which is an integer as required by constraint (C). When using the SPW iteration macro to iterate over N_s, N must be an integer multiple of the least common multiple of all the values of N_s. This insures that constraint (C) is satisfied throughout the iteration.

The additional N term in constraint (B) is a result of buffering requirements. If the number of iterations for each simulation are chosen to be greater than the required number I_Q, more than k P_b values may be computed. Conversely, if the number of iterations is less than required, fewer than k P_b values will be computed. ***For the Spread Spectrum systems in this report, the required number of iterations is $I_S = I_Q + N_s$,*** due to time delay in the PSK demodulator block.

Note: The CPA with Gaussian Tails nonlinearities (CPA/PDF_GT_NONLI and CPA/PDF_GT_NL_NC) must have at least four bins to function properly.

Fundamental Constraint Box (2)

5.2 Detailed Parameter Discussion

This following discussion outlines the reasons why the parameters in Table 3 are the preferred way to specify the configuration of the SPW systems. In addition, two very important relationships which are summarized below are discussed:

Maintain $R_s = f_s / N_s$ at all times
For the SS systems, also maintain $N_s = N_c PG$ at all times

Fundamental Constraint Box (3)

The derivations in the following sections are performed in each channel separately. QPSK modulation may be viewed as Binary Phase Shift Keying (BPSK) in the In-Phase and Quadrature channels; each QPSK symbol consists of two BPSK bits. The Continuous Wave (CW) jammer may be viewed as a sinusoid in each channel, and the Partial Band (PB) jammer may be viewed as a filtered Gaussian signal in each channel. E_b/N_0 and J/S are the same for each channel and for the composite signal. In addition, the bit rate of each channel, R_b, is equal to R_s, and the number of samples per bit, N_b, is equal to N_s.

5.2.1 System Parameters Related to the Information Signal and Gaussian Noise

<u>Signal Energy</u>

The energy, E, of a signal $s(t)$ is defined as

$$E \triangleq \int_{-\infty}^{\infty} |s(t)|^2 dt \qquad (3)$$

For discrete signals this becomes

$$E = \sum_{n=-\infty}^{\infty} |s(n)|^2 \frac{1}{f_s} \tag{4}$$

where f_s is the sampling frequency. A BPSK bit maintains a constant level $\pm A$ for the entire bit period. The energy of one bit, E_b, is given by

$$E_b = \sum_{n=0}^{N_s} \frac{A^2}{f_s} = \frac{A^2 N_s}{f_s} = \frac{A^2}{R_s} \tag{5}$$

where N_s is equal to the number of samples per bit and $R_s = \frac{f_s}{N_s}$ is equal to the bit rate.

In many analyses it is desirable to compute P_b as a function of E_b/N_0, where N_0 is the background noise power. This can be achieved by making E_b/N_0 a parameter, $\{E_b/N_0\}$, and making N_0 a parameter, $\{N_0\}$. Thus, for BPSK

$$\{E_b/N_0\} = \frac{E_b}{\{N_0\}} = \frac{A^2}{\{N_0\}R_s} \tag{6}$$

Then the bit amplitude is

$$A^2 = \{E_b/N_0\} \frac{\{N_0\} f_s}{N_s} \tag{7}$$

Signal Power

The power, P, of a signal is defined as

$$P \triangleq \frac{1}{T} \int_T |s(t)|^2 dt \tag{8}$$

where T is the period of the signal. For discrete signals the power is

$$P = \frac{1}{T} \sum_{T} |s(n)|^2 \frac{1}{f_s} \tag{9}$$

For a BPSK signal with amplitude $\pm A$ the power, S, is

$$S = \frac{f_s}{N_s} \sum_{n=0}^{N_s} \frac{A^2}{f_s} = A^2 \tag{10}$$

5.2.2 System Parameters Relating the Continuous Wave Jammer to the Information Signal

The power of a CW jammer is

$$J = \frac{1}{2\pi} \int_{-\pi}^{\pi} A_j^2 \cos^2(\omega_j t) dt = \frac{A_j^2}{2} \tag{11}$$

where A_j is the jammer amplitude and ω_j is the jammer frequency. The jammer amplitude can be found as a function of the parameter $\{J/N_0\}$:

$$\{J/N_0\} = \frac{J}{\{N_0\}} = \frac{A_j^2/2}{\{N_0\}} \tag{12}$$

$$A_j^2 = 2\{J/N_0\}\{N_0\} \tag{13}$$

It is often more useful to write the jammer amplitude as a function of the parameter $\{J/S\}$.

$$\{J/S\} = \frac{J}{S} = \frac{A_j^2/2}{A^2} \tag{14}$$

$$A_j^2 = 2A^2\{J/S\} = 2\{E_b/N_0\}\frac{\{N_0\}f_s}{N_s}\{J/S\} \tag{15}$$

Relationship between CW Jammer Parameters

It is quite straightforward to show that

$$\{J/S\} = \frac{\dfrac{J}{\{N_0\}}}{\{E_b/N_0\}\dfrac{f_s}{N_s}} \tag{16}$$

Usually these parameters are specified in deciBels (dB), and the following relationship results:

$$\{J/S\}_{(dB)} = \{J/N_0\}_{(dB)} + 10\log_{10}\left[\frac{N_s}{\{E_b/N_0\}f_s}\right] \tag{17}$$

Note that in Eq. (17) $\{E_b/N_0\}$ is not in dB.

Choice of CW Jammer Parameters

One of the key assumptions in the LO derivation is that the signal is small compared to the jammer, i.e. the ratio A_j^2/A^2 is large. From Eqs. (7) and (13) it is seen that if the parameter $\{J/N_0\}$ is used, this ratio becomes

$$\frac{A_j^2}{A^2} = \frac{2\{J/N_0\}}{\{E_b/N_0\}\dfrac{f_s}{N_s}} \tag{18}$$

This means that the validity of the small signal assumption is not only dependent on $\{J/N_0\}$, but also on many other parameters. However, if the parameter $\{J/S\}$ is used instead of $\{J/N_0\}$, the ratio A_j^2/A^2 is determined from Eqs. (7) and (15) to be

$$\frac{A_j^2}{A^2} = \frac{2\{E_b/N_0\}\frac{\{N_0\}f_s}{N_s}\{J/S\}}{\{E_b/N_0\}\frac{\{N_0\}f_s}{N_s}} = 2\{J/S\} \tag{19}$$

Now the small signal assumption is only dependent on $\{J/S\}$. For this reason, the use of the $\{J/S\}$ parameter is preferred over $\{J/N_0\}$.

Frequency Parameters

The message bandwidth is proportional to the bit rate, and $R_s = \frac{f_s}{N_s}$. If N_s is varied, the message bandwidth will change but f_j, the jammer frequency, will not. Thus, the spectrum of the message sequence will expand or contract while the spectral position of the jammer will remain unchanged. As the relative position of the jammer and the message change, the performance of the system will change dramatically. In order to isolate the performance of the MNP from the effects of this shift, it is useful to use a frequency ratio, $\{f_j/R_s\}$, instead of $\{f_j\}$. Using this ratio, the jammer frequency is $f_j = \{f_j/R_s\}R_s$. This way f_j and the message bandwidth change in the same manner when R_s is varied.

Alternatively, the sampling frequency may be set to $f_s = \{N_s\}\{R_s\}$. Then if R_s is constant, the message bandwidth will remain constant as N_s is varied. The spectral positions of both the jammer and the message signal will remain constant.

5.2.3 System Parameters Relating the Partial Band Jammer to the Information Signal

PB Jammer Power

The autocorrelation function, $r_w(m)$, for sampled white Gaussian noise, $w(n)$, is given by

$$r_w(m) = \sigma_w^2 \delta(m) \tag{20}$$

where σ_w^2 is the variance of $w(n)$ and $\delta(m) = \begin{cases} 1, & m = 0 \\ 0, & else \end{cases}$ is the Kronecker delta function. The Power Spectral Density (PSD), $S_w(f)$, is given by

$$S_w(f) = \sigma_w^2 \quad \text{for all } f \tag{21}$$

The frequency response of an ideal lowpass filter is

$$H(f) = \begin{cases} 1, & |f| < f_c \\ 0, & f_c < |f| < \frac{1}{2} \end{cases} \tag{22}$$

where f_c is the cutoff frequency (normalized by the sampling frequency, f_s). The Gaussian noise PB jammer is constructed by passing white Gaussian noise through the lowpass filter (for simulation at baseband). Therefore, the PSD of the PB jammer, $S_{PB}(f)$, is

$$S_{PB}(f) = \begin{cases} \sigma_w^2, & |f| < f_c \\ 0, & f_c < |f| < \frac{1}{2} \end{cases} \tag{23}$$

The power of the PB jammer, J_{PB}, may be found by integrating the PSD over the domain of f, resulting in

$$J_{PB} = \int_{-1/2}^{1/2} S_{PB}(e^{j2\pi f})\,df = 2f_c\sigma_w^2 \tag{24}$$

System Parameters

Given the jammer-to-signal ratio parameter $\{J_{PB}/S\}$, then

$$\{J_{PB}/S\} = \frac{J_{PB}}{A^2} = \frac{J_{PB}}{\{E_b/N_0\}\dfrac{\{N_0\}f_s}{N_s}} = \frac{2f_c N_s \sigma_w^2}{\{E_b/N_0\}\{N_0\}f_s} \tag{25}$$

To find the required value of σ_w^2 for a given $\{J_{PB}/S\}$

$$\sigma_w^2 = \frac{\{J_{PB}/S\}\{E_b/N_0\}\{N_0\}}{2f_c}\frac{f_s}{N_s} = \frac{\{J_{PB}/S\}\{E_b/N_0\}\{N_0\}}{2f_c}R_s \tag{26}$$

If f_c is chosen as a fraction of the bit rate, i.e. if $\{f_c/R_s\}$ is used then

$$\sigma_w^2 = \frac{\{J_{PB}/S\}\{E_b/N_0\}\{N_0\}}{2\{f_c/R_s\}} \tag{27}$$

5.2.4 Spread Spectrum System Parameters

The processing gain, number of samples per chip, and number of samples per symbol are related by

$$N_s = N_c PG \tag{28}$$

Also, since $R_s = \frac{f_s}{N_s}$, then the chip rate, R_c, is equal to

$$R_c = \frac{f_s}{N_c} = R_s \frac{N_s}{N_c} = R_s PG \tag{29}$$

6. References

[Comd91] Comdisco Systems, Inc., SPW Version 2.8 User's Guide, May 1991.

[Grim93] Grimm, Jimm H. et. al., "Continuous Polynomial Approximation", to be published in *Proceedings of MILCOM '93*, October 1993.

[Illi91] Illinois Institute of Technology Report, Continuation Study of a Communications Receiver for Spread Spectrum Signals, RADC Contract No. F49620-88-C-0053/SB5881-0378, January 31, 1991.

* [Illi93a] Illinois Institute of Technology Report, A Spread Spectrum Communications Receiver with Nonlinear Processing, RL Contract No. F30602-91-C-0059, RL-TR-93-50, May 93, ADB174588.

[Illi93b] Illinois Institute of Technology Report, Investigation and Simulation of Nonlinear Processors for Spread Spectrum Receivers Volume I, RL Contract No. F30602-92-C-0039, RL-TR-93-258, Vol I, Dec 93.

[Taub86] Taub, Herbert and Schilling, Donald L., Principles of Communication Systems 2nd edition, McGraw Hill, 1986

[Varn93] Varn, David, "Automatic Parameter Iteration in SPW", *WaveForum* by Comdisco Systems, Inc., March 1993.

SIM/PE

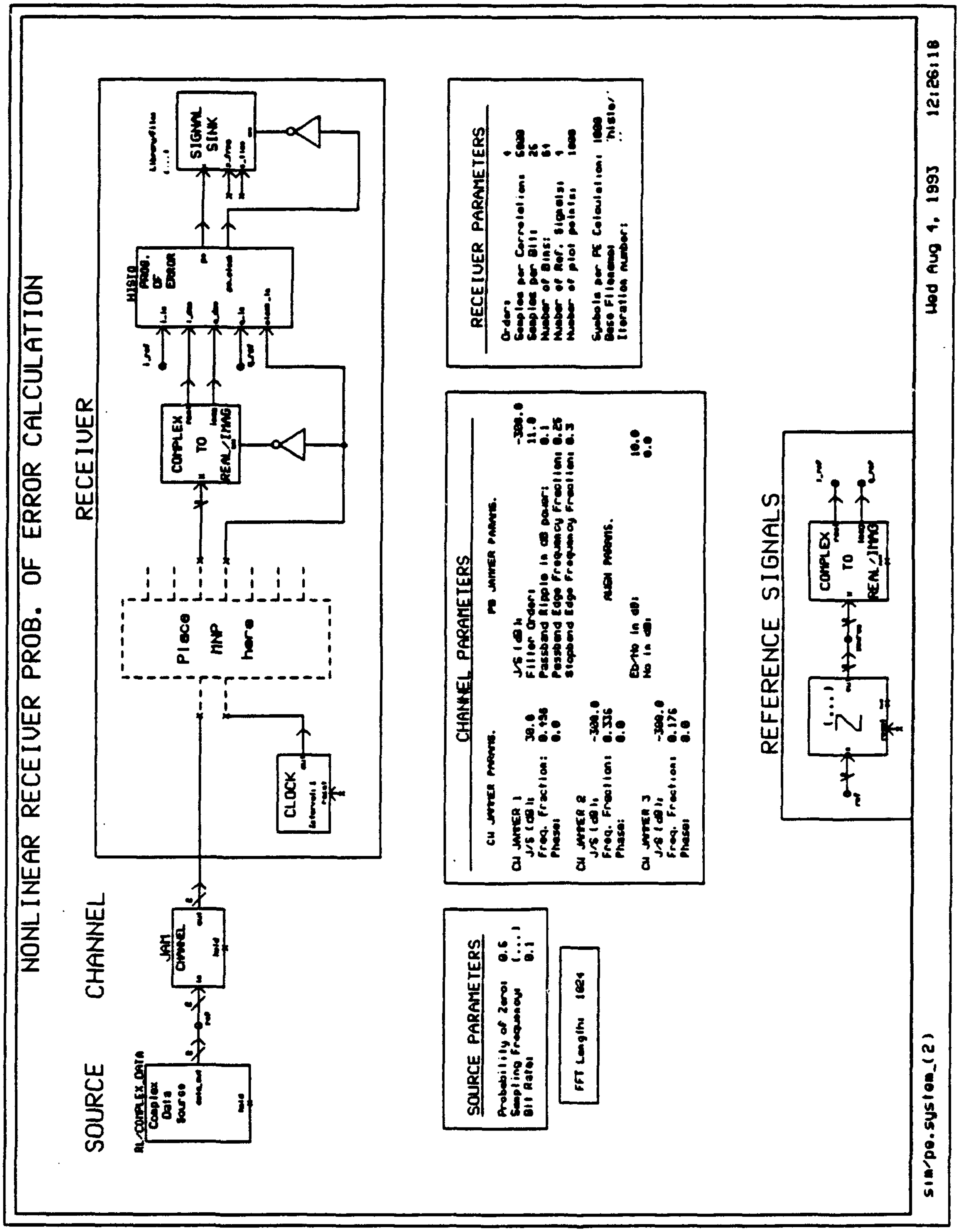


Figure (1)

SIM/PES

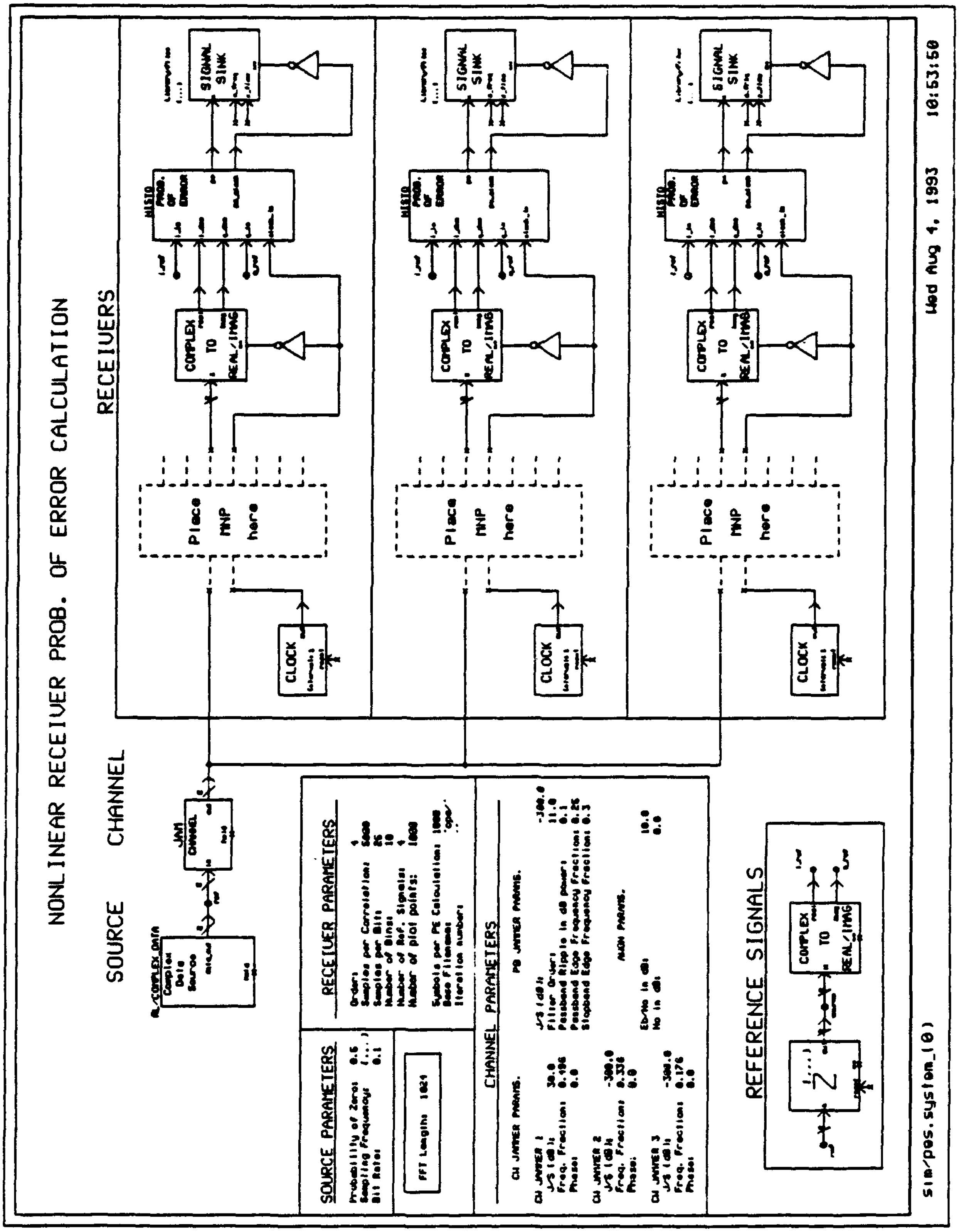


Figure (2)

SIM/HSIGS

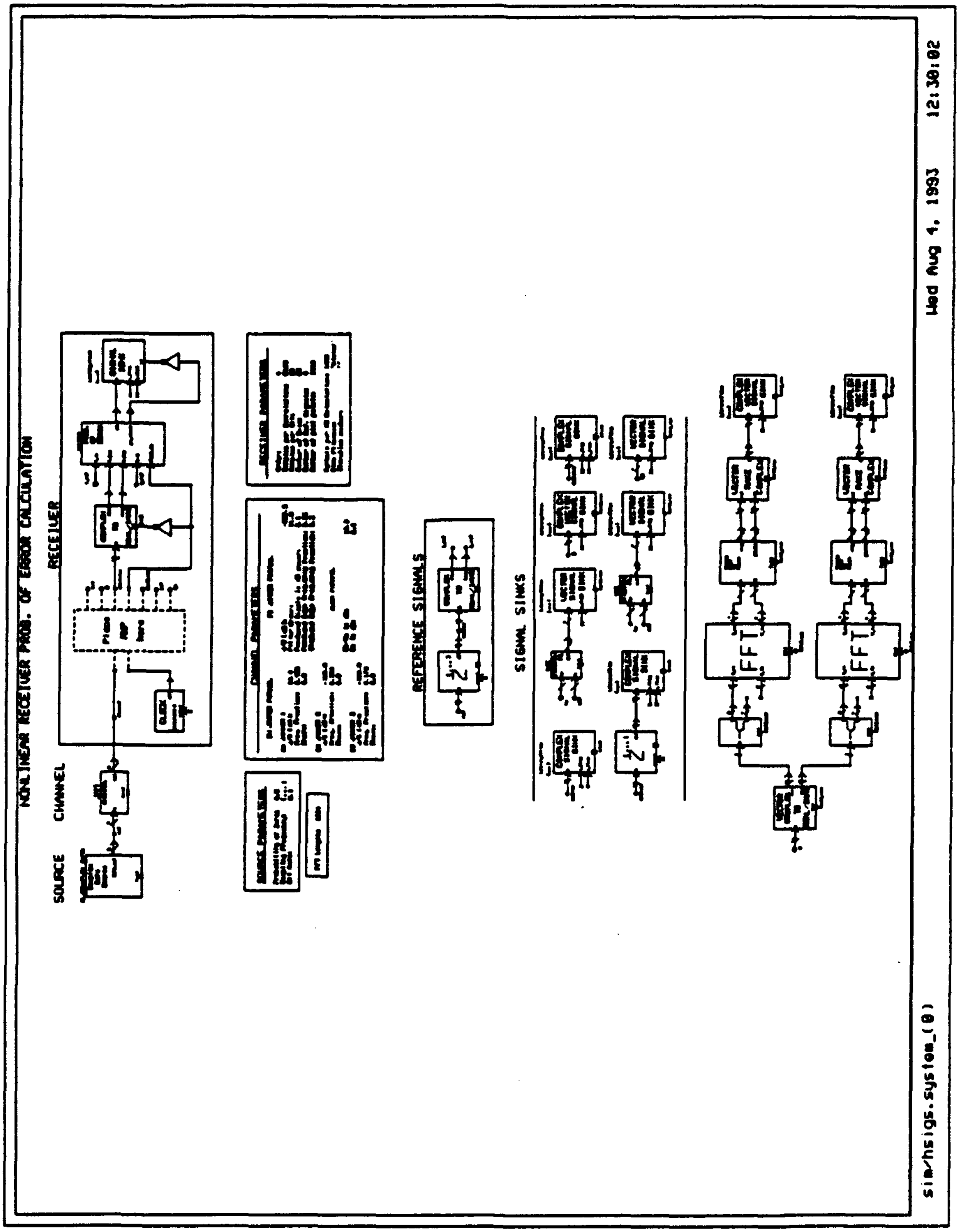


Figure (3)

SIM/PSIGS

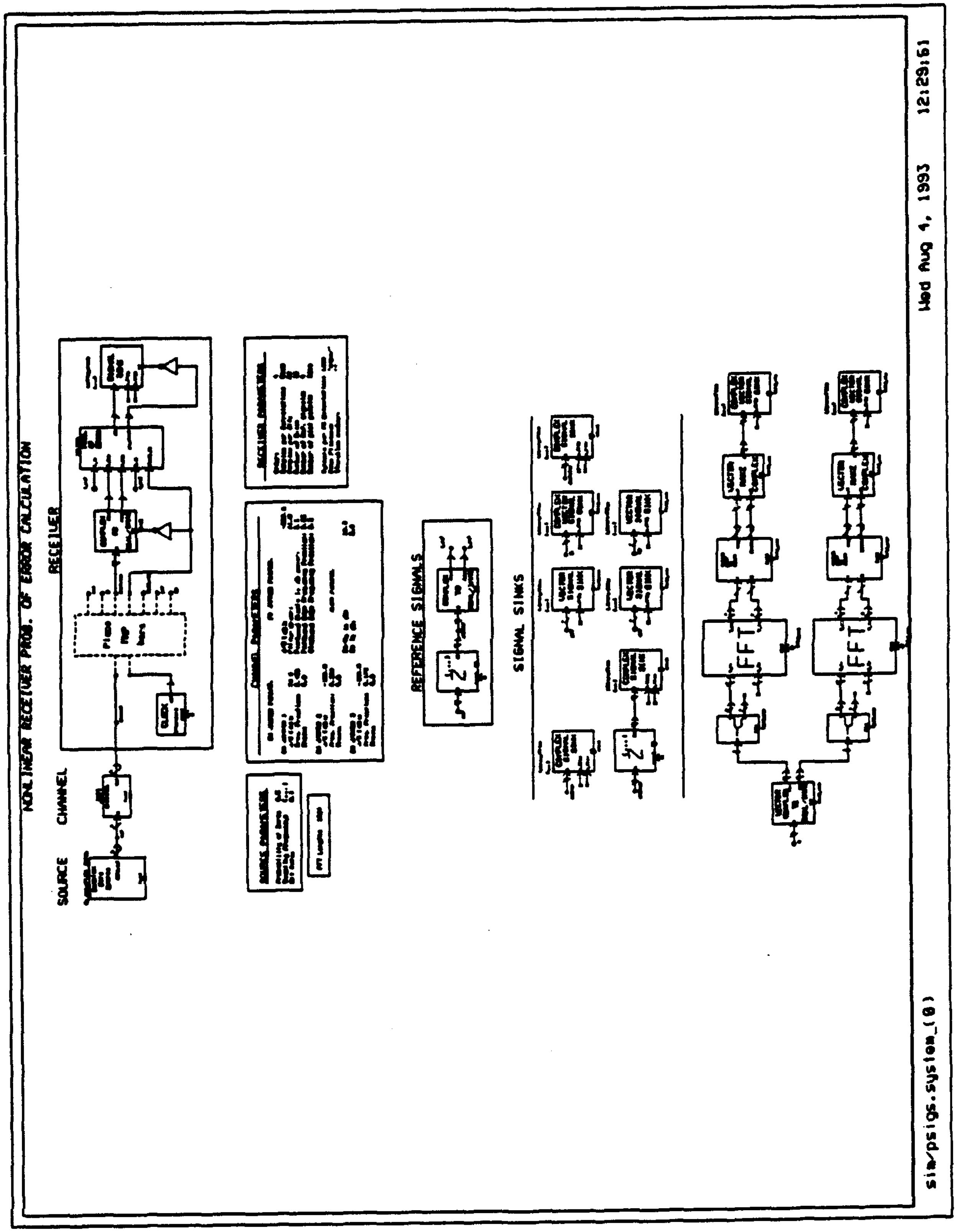


Figure (4)

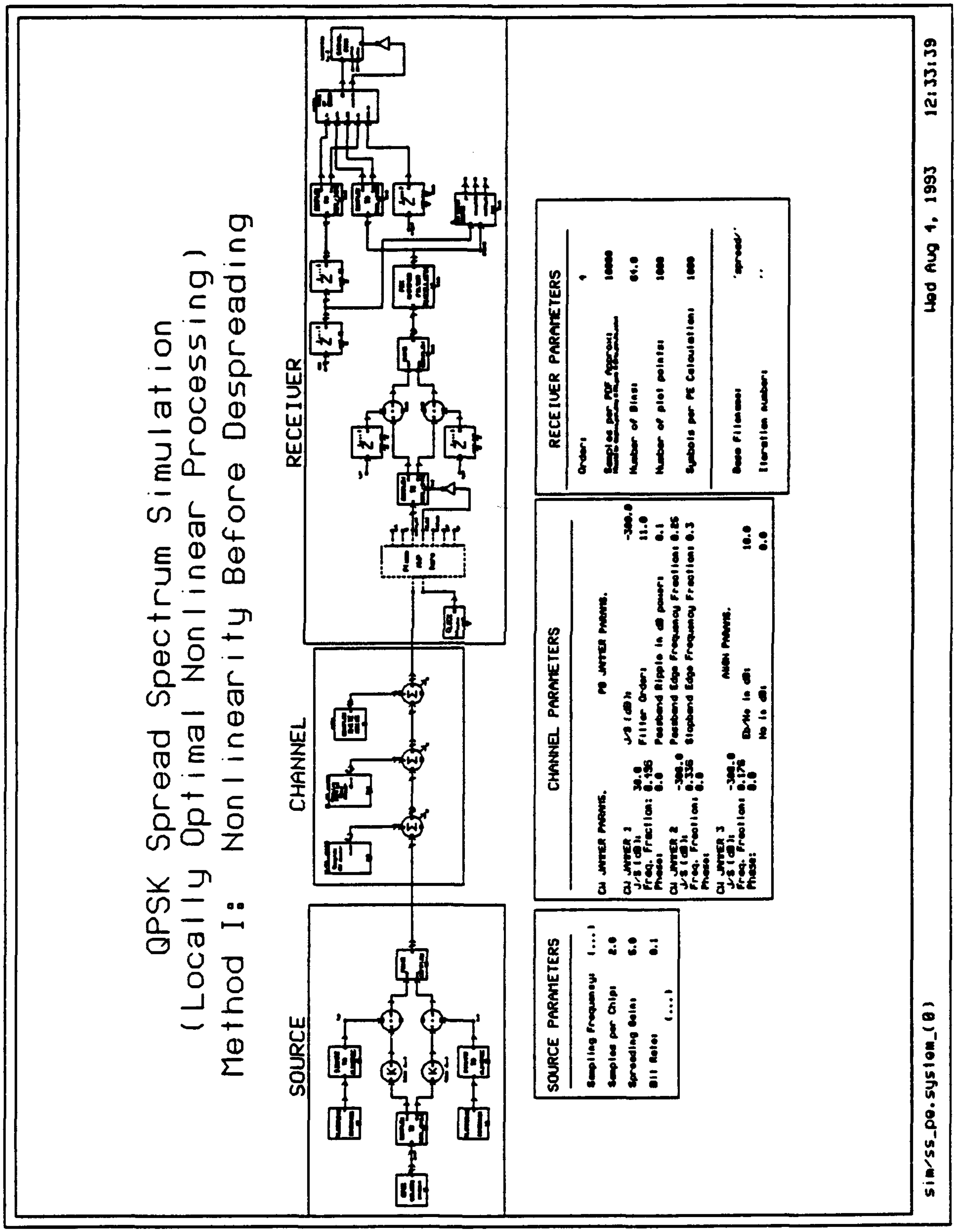

QPSK Spread Spectrum Simulation
(Locally Optimal Nonlinear Processing)
Method I: Nonlinearity Before Despreading
SOURCE
CHANNEL
RECEIVER
SOURCE PARAMETERS
CHANNEL PARAMETERS
RECEIVER PARAMETERS
sim/ss_pe.system_(0)
Wed Aug 4, 1993
12:33:39

Figure (5)

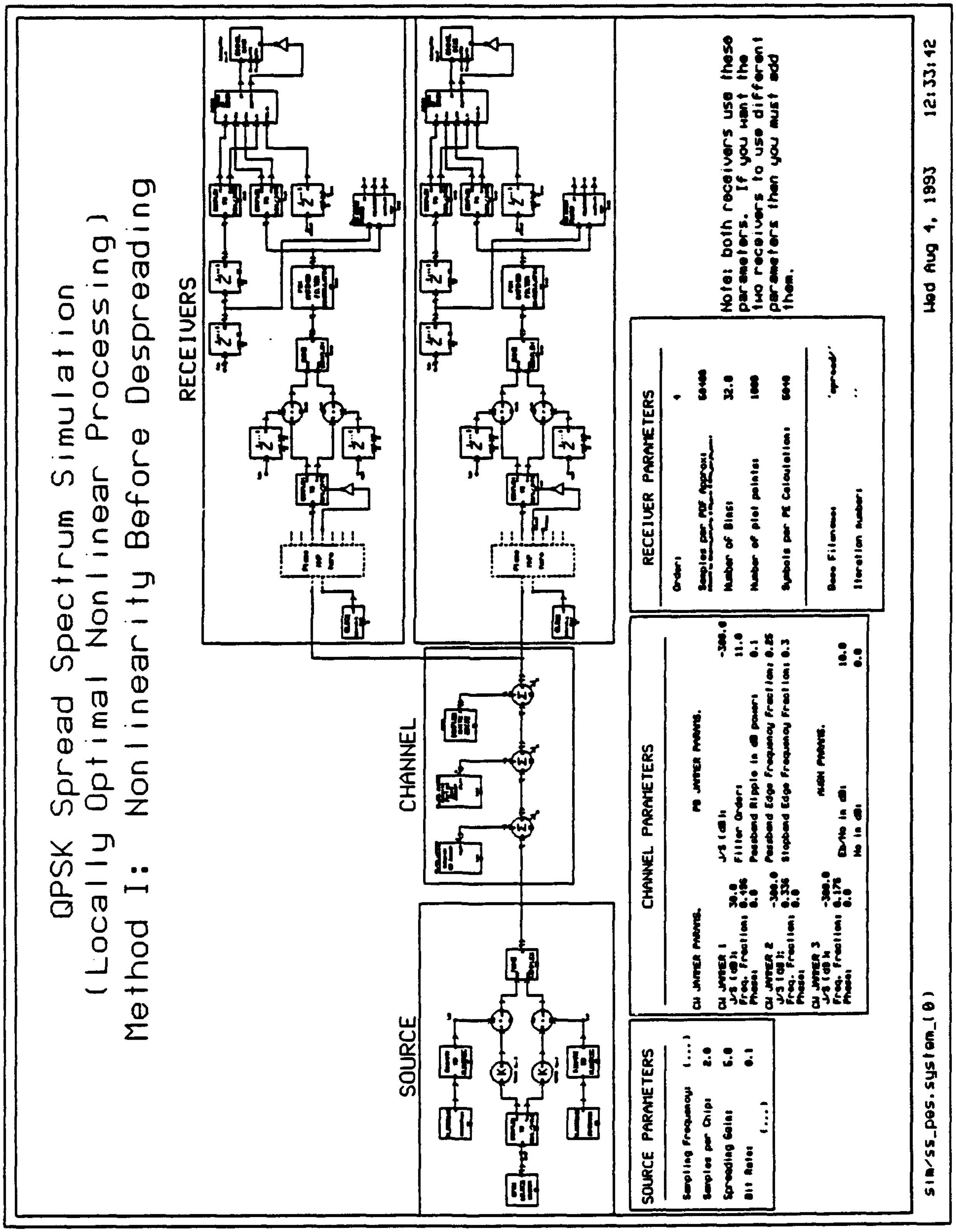

QPSK Spread Spectrum Simulation
(Locally Optimal Nonlinear Processing)
Method I: Nonlinearity Before Despreading
SOURCE
CHANNEL
RECEIVERS
SOURCE PARAMETERS
CHANNEL PARAMETERS
RECEIVER PARAMETERS
Note: both receivers use these parameters. If you want the two receivers to use different parameters then you must add them.
Wed Aug 4, 1993
12:33:42
sim/ss_pes.system_(0)

Figure (6)

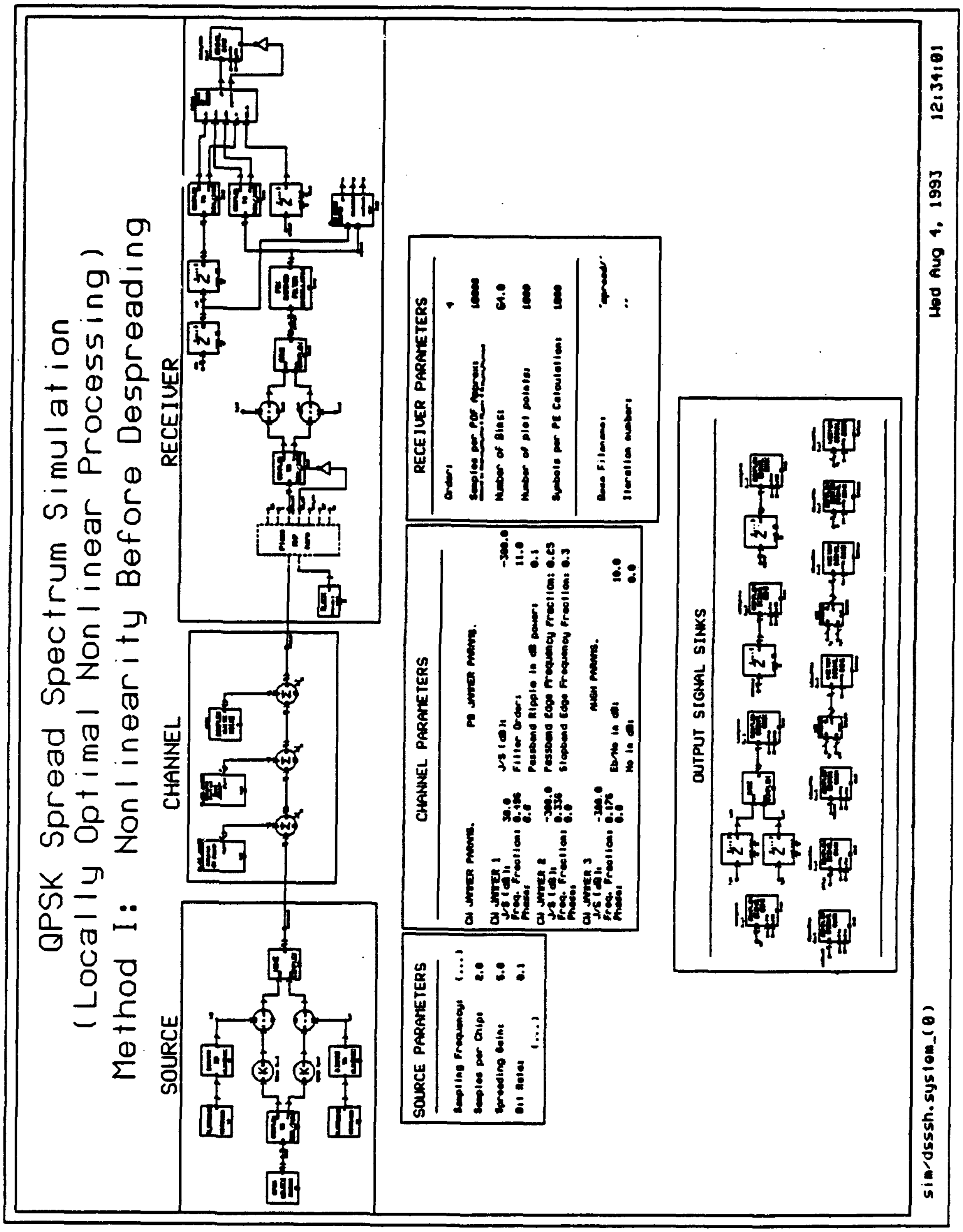

QPSK Spread Spectrum Simulation
(Locally Optimal Nonlinear Processing)
Method I: Nonlinearity Before Despreading
SOURCE
CHANNEL
RECEIVER
SOURCE PARAMETERS
CHANNEL PARAMETERS
RECEIVER PARAMETERS
OUTPUT SIGNAL SINKS
sim/dsssh.system_(0)
Wed Aug 4, 1993
12:34:01

Figure (7)

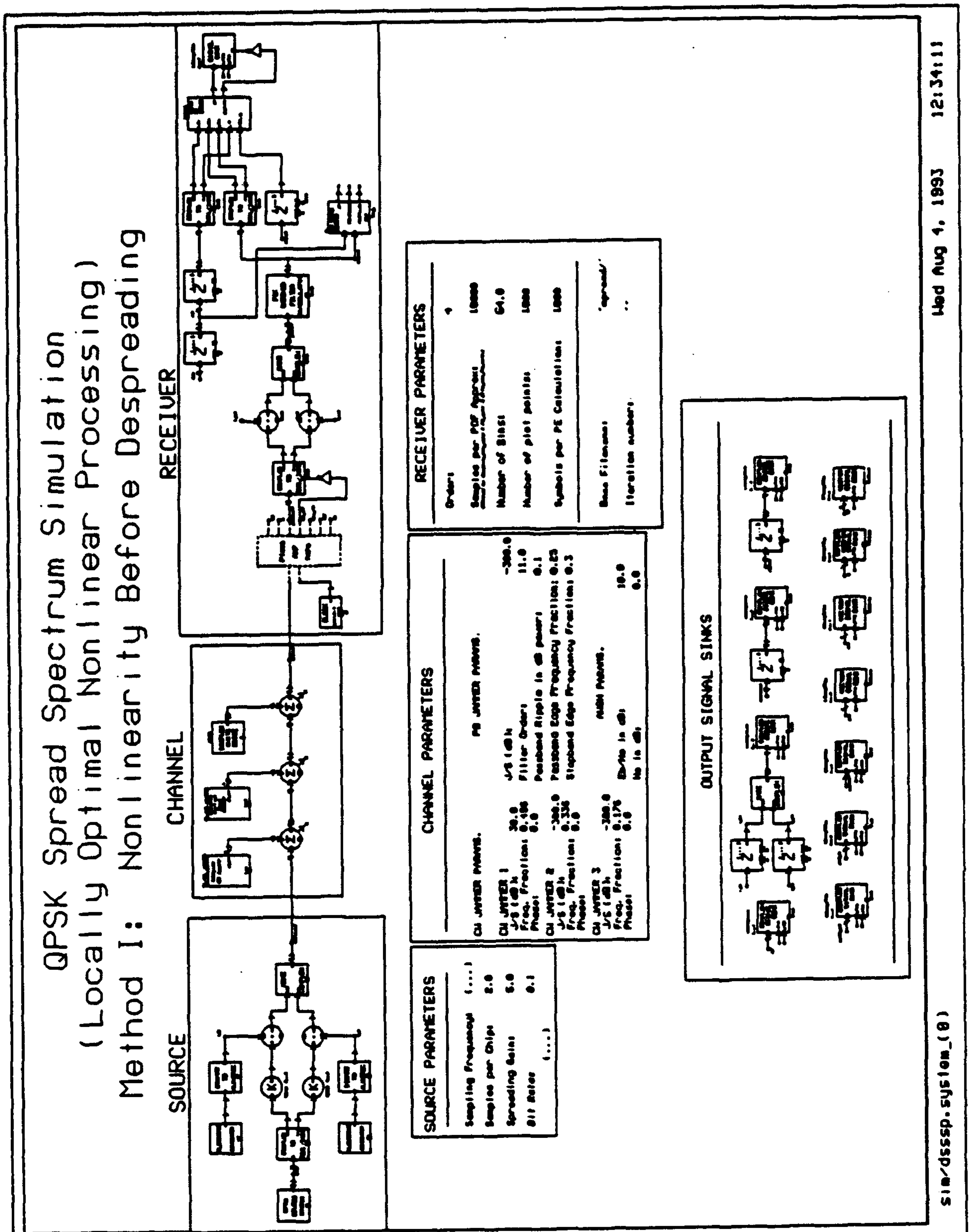

QPSK Spread Spectrum Simulation
(Locally Optimal Nonlinear Processing)
Method I: Nonlinearity Before Despreading
SOURCE
CHANNEL
RECEIVER
SOURCE PARAMETERS
CHANNEL PARAMETERS
RECEIVER PARAMETERS
OUTPUT SIGNAL SINKS
sim/dsssp.system_(8)
Wed Aug 4, 1993
12:34:11

Figure (8)

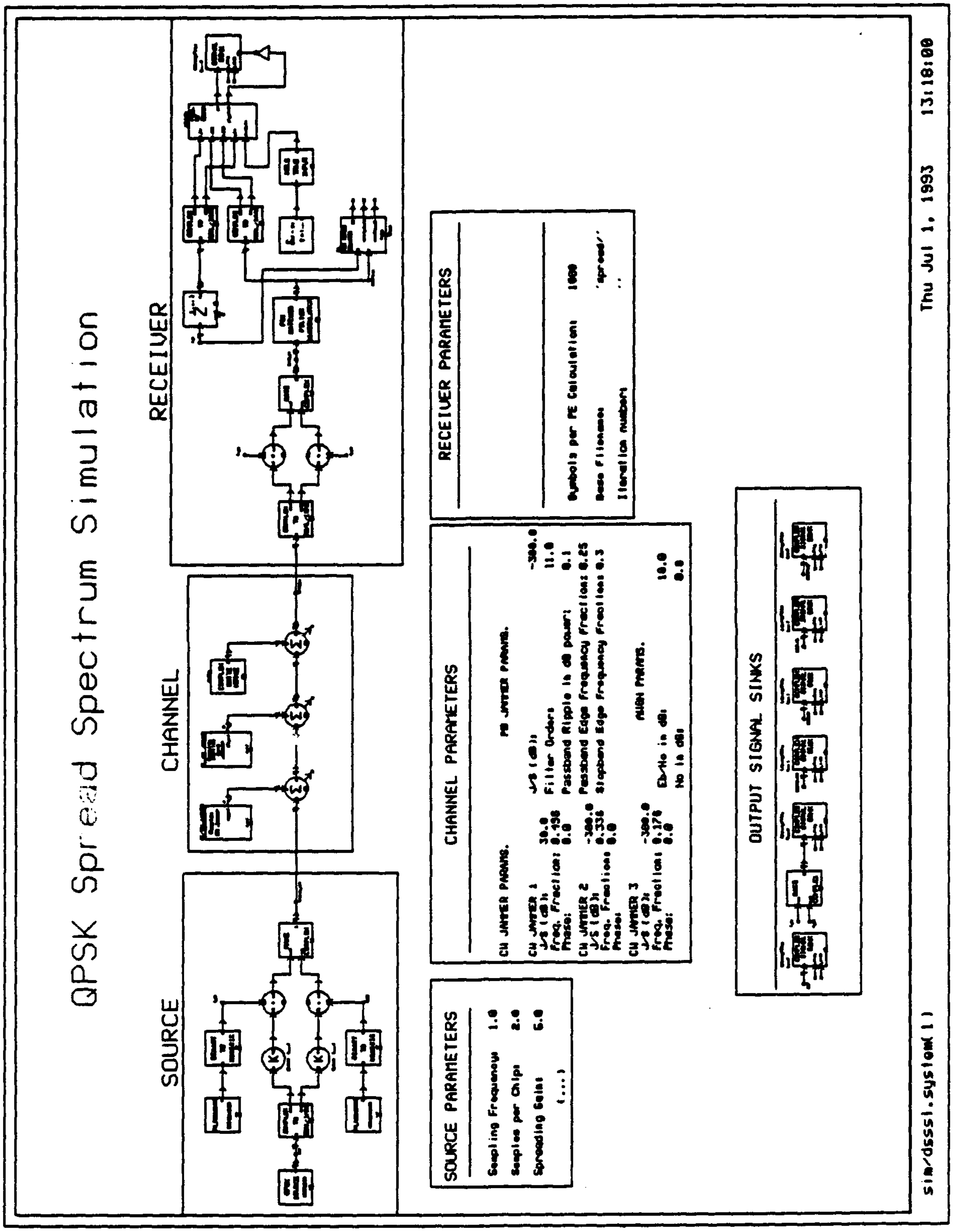

QPSK Spread Spectrum Simulation
SOURCE
CHANNEL
RECEIVER
SOURCE PARAMETERS
Sampling Frequency 1.0
Samples per Chips 2.0
Spreading Gain 6.0
CHANNEL PARAMETERS
RECEIVER PARAMETERS
OUTPUT SIGNAL SINKS
Thu Jul 1, 1993 13:18:00
sim/dsssl.system(1)

Figure (9)

Appendix A

Docgen Listing

This Appendix is an alphabetical listing of the help screens available for each IIT SPW block. In this Appendix the inputs, parameters and outputs of each block are capitalized in the description of the block to distinguish them from the text. However, in the listing of the inputs, parameters and outputs, the capitalization matches the capitalization of the actual parameter names in SPW.

The names of hierarchical blocks are marked with a dagger †. The details of these blocks are printed in alphabetical order in Appendix B.

Name:
cpa/cdf2†

Description:
This hierarchical block generates a Cumulative Distribution Function (CDF) and a Probability Density Function (PDF) of the magnitude of the input message by applying the Continuous Polynomial Approximation (CPA) algorithm to a histogram CDF. A vec/heap_sort block is used to sort the data to prepare it for the histo/equi block, which generates the histogram CDF. The cpa/slope1 block computes the derivatives of the histogram needed by the CPA algorithm. The cpa/coef_lt block uses the equiprobable bin histogram CDF, its derivatives, and the breakpoints to compute the CPA CDF and PDF.

This block applies the linear transform to the polynomial. Without the linear transform the magnitude values in each bin would range from BP[K-1] to BP[K]. The linear transform shifts each bin to the origin, so the new magnitude values range from 0 to BP[K]-BP[K-1].

Inputs:

data	Magnitude of input message sequence

Parameters:

bins	Number of bins
samples	Number of samples

Outputs:

bp	Breakpoints (CDF and PDF interval boundaries)
cdf	CDF polynomial coefficients
pdf	PDF polynomial coefficients

See also:
vec/heap_sort, histo/equi, cpa/slope1, cpa/coef_lt

Name:
cpa/cdf2_nonlin†

Description:
This is a hierarchical block which applies the nonlinearity to the input signal in vector format. The histo/rec_to_polar block converts the two dimensional input signal from rectangular to polar coordinates, and a CPA of the Probability Density Function (PDF) of the magnitude is computed by the cpa/cdf2 block. The poly/mnt block obtains the Memoryless Nonlinear Transform

(MNT) from the CPA PDF and applies it to the magnitude of the input message sequence. The result is converted back to rectangular coordinates by the histo/polar_to_rec block using the unmodified phase. The poly/plot block generates a plot of the PDF and CDF from their polynomial coefficients.

The coefficients in the cpa/cdf2 block are computed with the linear transform applied, thus each bin is shifted to the origin.

Refer to the poly/mnt block for more information on the MNT.

Inputs:

i_in	In-Phase component of input message sequence
q_in	Quadrature component of input message sequence

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram PDF
points	Number of plot points for PDF and MNT

Outputs:

pdf	CPA PDF
i_out	In-Phase component of transformed sequence
q_out	Quadrature component of transformed sequence
mnt	Plot of the Memoryless Nonlinear Transform
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/rec_to_polar, cpa/cdf2, poly/mnt, poly/plot, histo/polar_to_rec

---

Name:
cpa/cdf3†

Description:
This hierarchical block generates a Cumulative Distribution Function (CDF) and a Probability Density Function (PDF) of the magnitude of the input message by applying the Continuous Polynomial Approximation (CPA) algorithm to a histogram CDF. A vec/heap_sort block is used to sort the data to prepare it for the histo/equi block, which generates the histogram CDF. The cpa/slope1 block computes the derivatives of the histogram needed by the CPA algorithm. The

cpa/coef_ltdaf block uses the equiprobable bin histogram CDF, its derivatives, and the breakpoints to compute the CPA CDF and PDF.

This block (cpa/cdf3) is identical to the cpa/cdf2 block except that the constraint that the CDF be continuous has been relaxed. Thus, the equations to determine the coefficient values are different.

Like the cpa/cdf2 block, this block applies the linear transform to the polynomial. Without the linear transform the magnitude values in each bin would range from BP[K-1] to BP[K]. The linear transform shifts each bin to the origin, so the new magnitude values range from 0 to BP[K]-BP[K-1].

Inputs:
  data — Magnitude of input message sequence

Parameters:
  bins — Number of bins
  samples — Number of samples

Outputs:
  bp — Breakpoints (CDF and PDF interval boundaries)
  cdf — CDF polynomial coefficients
  pdf — PDF polynomial coefficients

See also:
  vec/heap_sort, histo/equi, cpa/slope1, cpa/coef_ltdaf

Name:
  cpa/cdf3_nonlin†

Description:
  This is a hierarchical block which applies the nonlinearity to the input signal in vector format. The histo/rec_to_polar block converts the two dimensional input signal from rectangular to polar coordinates, and a CPA of the Probability Density Function (PDF) of the magnitude is computed by the cpa/cdf3 block. The poly/mnt block obtains the Memoryless Nonlinear Transform (MNT) from the CPA PDF and applies it to the magnitude of the input message sequence. The result is converted back to rectangular coordinates by the histo/polar_to_rec block using the

unmodified phase. The poly/plot block generates a plot of the PDF and CDF from their polynomial coefficients.

The coefficients in the cpa/cdf3 block are computed with the linear transform applied, thus each bin is shifted to the origin. Also, the constraint that the CDF be continuous is removed.

Refer to the poly/mnt block for more information on the MNT.

Inputs:

i_in	In-Phase component of input message sequence
q_in	Quadrature component of input message sequence

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram PDF
points	Number of plot points for PDF and MNT

Outputs:

pdf	CPA PDF
i_out	In-Phase component of transformed sequence
q_out	Quadrature component of transformed sequence
mnt	Plot of the Memoryless Nonlinear Transform
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/rec_to_polar, cpa/cdf3, poly/mnt, poly/plot, histo/polar_to_rec

---

Name:
cpa/coef_gt

Description:
This block computes the Continuous Polynomial Approximation (CPA)to a function. In this implementation, the CPA is an approximation of a Probability Density Function (PDF).

The coefficients in this routine are computed with the linear transform applied, thus each bin is shifted to the origin. The first and last bins are a Gaussian tail instead of a polynomial. The mean and variance for the Gaussian are taken to be the mean and variance of the received signal.

Note that the value for BINS must be set to a value greater than or equal to 4.

WARNING:
This program modifies the histogram PDF input. If another block is going to also use the histogram PDF results will be unpredictable.

Inputs:

bp	Breakpoints (PDF interval boundaries)
f	Histogram PDF
df	First derivative of Histogram PDF
mean	Average of the received signal
var	Variance of the received signal

Parameters:

bins	Number of bins

Outputs:

pdf	CPA PDF

Name:
cpa/coef_lt

Description:
This block computes the Continuous Polynomial Approximation (CPA)to a function. In this implementation, the CPA is an approximation of a Cumulative Distribution Function (CDF).

The coefficients in this routine are computed with the linear transform applied, thus each bin is shifted to the origin.

Inputs:

bp	Breakpoints (CDF and PDF interval boundaries)
histo_cdf	Histogram CDF
df	First derivative of Histogram CDF
ddf	Second derivative of Histogram CDF

Parameters:

bins	Number of bins

Outputs:

cdf	CPA CDF
pdf	CPA PDF

Name:
cpa/coef_ltdaf

Description:
This block computes the Continuous Polynomial Approximation (CPA)to a function. In this implementation, the CPA is an approximation of a Cumulative Distribution Function (CDF).

The coefficients in this routine are computed with the linear transform applied, thus each bin is shifted to the origin. Also, the constraint that the CDF be continuous is removed.

Inputs:

bp	Breakpoints (CDF and PDF interval boundaries)
df	First derivative of Histogram CDF
ddf	Second derivative of Histogram CDF

Parameters:

bins	Number of bins

Outputs:

cdf	CPA CDF
pdf	CPA PDF

Name:
cpa/pdf_gt†

Description:
This hierarchical block generates a Probability Density Function (PDF) of the magnitude of the input message by applying the Continuous Polynomial Approximation (CPA) algorithm to a histogram PDF. A vec/heap_sort block is used to sort the data to prepare it for the histo/equi block, which generates the histogram PDF. The vec/ave_bp obtains values of the histogram at the breakpoints by averaging the values of the adjacent bins. The cpa/slope2 block computes the derivatives of the histogram needed by the CPA algorithm. The cpa/coef_gt block uses the equiprobable bin histogram, its derivatives, the breakpoints, and the mean and variance of the magnitude to compute the CPA PDF.

The first and last bins are approximated using Gaussian tails instead of polynomials. Because of this, the poly/mnt_gt and poly/plot_gt blocks should be used with this block.

The linear transform is applied to the polynomial bins. Without the linear transform the magnitude values in each bin would range from BP[K-1] to BP[K]. The linear transform shifts each bin to the origin, so the new magnitude values range from 0 to BP[K]-BP[K-1].

Note that the value for BINS must be set to a value greater than or equal to 4.

Inputs:

data	Magnitude of input message sequence

Parameters:

bins	Number of bins
samples	Number of samples

Outputs:

bp	Breakpoints (PDF interval boundaries)
pdf	PDF polynomial coefficients

See also:
vec/heap_sort, histo/equi, cpa/slope2, cpa/coef_ltdaf, poly/mnt_gt, poly/plot_gt

---

Name:
cpa/pdf_gt_nonlin†

Description:
This is a hierarchical block which applies the nonlinearity to the input signal in vector format. The histo/rec_to_polar block converts the two dimensional input signal from rectangular to polar coordinates, and a CPA of the Probability Density Function (PDF) of the magnitude is computed by the cpa/pdf_gt block. The poly/mnt block obtains the Memoryless Nonlinear Transform (MNT) from the CPA PDF and applies it to the magnitude of the input message sequence. The result is converted back to rectangular coordinates by the histo/polar_to_rec block using the unmodified phase. The poly/plot block generates a plot of the PDF and CDF from their polynomial coefficients.

The coefficients in the cpa/pdf_gt block are computed with the linear transform applied, thus each bin is shifted to the origin. The first and last bins are a Gaussian tail instead of a polynomial. The mean and variance for the Gaussian are taken to be the mean and variance of the received signal.

Refer to the poly/mnt block for more information on the MNT.

Inputs:

i_in	In-Phase component of input message sequence
q_in	Quadrature component of input message sequence

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram PDF
points	Number of plot points for PDF and MNT

Outputs:

pdf	CPA PDF
i_out	In-Phase component of transformed sequence
q_out	Quadrature component of transformed sequence
mnt	Plot of the Memoryless Nonlinear Transform
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/rec_to_polar, cpa/pdf_gt, poly/mnt, poly/plot, histo/polar_to_rec

---

Name:
cpa/s_cdf2_nl_nc†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the cpa/cdf2_nonlin nonlinearity. This block does not generate the summation part of the LO decision statistic.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the cpa/s_cdf2_nl_nc block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the cpa/s_cdf2_nl_nc block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence (vector)
out	Complex transformed sequence (serial)
clk_out	Timing signal: goes high when valid data sample is available at the output of cpa/s_cdf2_nl_nc
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
cpa/cdf2_nonlin, romelib/timing, rl/inf_vsource, cpa/s_cdf2_nonlin

Name:
cpa/s_cdf2_nonlin†

Description:

This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the cpa/cdf2_nonlin nonlinearity. In addition, the summation in the LO decision statistic is formed using the histo/correlator2 block. The output of the correlator is converted back to serial form.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the cpa/s_cdf2_nonlin block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the cpa/s_cdf2_nonlin block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence
out	Decision of nonlinearity
clk_out	Timing signal: goes high when valid data sample is available at the output of cpa/s_cdf2_nonlin
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
cpa/cdf2_nonlin, histo/correlator2, romelib/timing, rl/inf_vsource

---

Name:
cpa/s_cdf3_nl_nc†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the cpa/cdf3_nonlin nonlinearity. This block does not generate the summation part of the LO decision statistic.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the cpa/s_cdf3_nl_nc block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the cpa/s_cdf3_nl_nc block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)&

"spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence (vector)
out	Complex transformed sequence (serial)
clk_out	Timing signal: goes high when valid data sample is available at the output of cpa/s_cdf3_nl_nc
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
cpa/cdf3_nonlin, romelib/timing, rl/inf_vsource, cpa/s_cdf3_nonlin

Name:
cpa/s_cdf3_nonlin†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the cpa/cdf3_nonlin nonlinearity. In addition, the summation in the LO decision statistic is formed using the histo/correlator2 block. The output of the correlator is converted back to serial form.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the cpa/s_cdf3_nonlin block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the cpa/s_cdf3_nonlin block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence
out	Decision of nonlinearity
clk_out	Timing signal: goes high when valid data sample is available at the output of cpa/s_cdf3_nonlin
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
cpa/cdf3_nonlin, histo/correlator2, romelib/timing, rl/inf_vsource

---

Name:
cpa/s_pdf_gt_nl_nc†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the cpa/pdf_gt_nonlin nonlinearity. This block does not generate the summation part of the LO decision statistic.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the cpa/s_pdf_gt_nl_nc block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the cpa/s_pdf_gt_nl_nc block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)&

"spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence (vector)
out	Complex transformed sequence (serial)
clk_out	Timing signal: goes high when valid data sample is available at the output of cpa/s_pdf_gt_nl_nc
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
cpa/pdf_gt_nonlin, histo/correlator2, romelib/timing, cpa/s_pdf_gt_nonli, rl/inf_vsource

Name:
cpa/s_pdf_gt_nonli†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the cpa/pdf_gt_nonlin nonlinearity. In addition, the summation in the LO decision statistic is formed using the histo/correlator2 block. The output of the correlator is converted back to serial form.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the cpa/s_pdf_gt_nonli block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the cpa/s_pdf_gt_nonli block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence
out	Decision of nonlinearity
clk_out	Timing signal: goes high when valid data sample is available at the output of cpa/s_pdf_gt_nonli
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
cpa/pdf_gt_nonlin, histo/correlator2, romelib/timing, rl/inf_vsource

---

Name:
cpa/slope1

Description:
This block computes the three point derivative of the histogram CDF for use in the CPA MNT construction. Extra bins are added to compute the slopes at the endpoints. This is the method described in Appendix 3 of IIT Final Report 1991, F30602-91-C-0059. This block is used in the cpa/cdf2 and cpa/cdf3 blocks.

Note:
The additional derivative at each end is always zero, since HISTO_CDF[-2]=HISTO_CDF[0]=0.0 and HISTO_CDF[BINS]=HISTO_CDF[BINS+2]=1.0.

Inputs:

bp	Breakpoints (CDF interval boundaries)
histo_cdf	Histogram CDF

Parameters:

bins	Number of bins

Outputs:

df	First derivative
ddf	Second derivative

Name:
cpa/slope2

Description:
This block computes the three point derivative of the histogram PDF for use in the CPA MNT construction. Extra bins are added to compute the slopes at the endpoints. This block is used in the cpa/pdf_gt block.

Note:
The additional derivative at each end is always zero, since HISTO_PDF[-2]=HISTO_PDF[0]=0.0andHISTO_PDF[BINS]=HISTO_PDF[BINS+2]=0.0.

Inputs:

bp	Breakpoints (PDF interval boundaries)
histo_pdf	Histogram PDF

Parameters:

bins	Number of bins

Outputs:

df	First derivative
ddf	Second derivative

Name:
fourier/mnt[†]

Description:
This block obtains the Memoryless Nonlinear Transform (MNT) from a Fourier Series Approximation (FSA) of a Probability Density Function (PDF). It is a hierarchical block which contains two fourier/mnt_calc blocks. One of them computes the MNT for each point of the magnitude of the input message sequence, and the other generates a plot of the MNT from the minimum to the maximum magnitude value with the aid of a vec/minmax_ramp block.

This block operates on the magnitude of the two dimensional message signal.

Inputs:

a	Vector of A coefficients (cosine terms)
b	Vector of B coefficients (sine terms)
T	"Period" of the FSA, equal to MAX - MIN (T is computed by the fourier/pdf block)
message	Input message sequence (2 dimensional)

Parameters:

samples	Number of samples in data vector
points	Number of plot points for MNT
P	Order of FSA

Outputs:

g_val	MNT of each data sample
mnt	Plot of MNT from minimum to maximum data value

See also:
fourier/mnt_calc, fourier/pdf, vec/minmax_ramp

Name:
fourier/mnt_calc

Description:
This block obtains the Memoryless Nonlinear Transform (MNT) from a Fourier Series Approximation (FSA) of a Probability Density Function (PDF). The MNT of each point of the input message sequence is stored in the G_VAL output.

This block operates on the magnitude of the two dimensional message signal.

Inputs:

a	Vector of A coefficients (cosine terms)
b	Vector of B coefficients (sine terms)
T	"Period" of the FSA, equal to MAX - MIN (T is computed by the fourier/pdf block)
message	Input message sequence (2 dimensional)

Parameters:

samples	Number of samples in data vector
P	Order of FSA

Outputs:

g_val	MNT of each data sample

Name:
fourier/nonlin†

Description:
This is a hierarchical block which applies the nonlinearity to the input signal in vector format. The histo/rec_to_polar block converts the two dimensional input signal from rectangular to polar coordinates, and a histogram approximation of the Probability Density Function (PDF) of the magnitude is computed by the histo/histo block. The fourier/pdf block then constructs a Fourier Series Approximation (FSA) of the PDF based on the histogram. The fourier/mnt block obtains the Memoryless Nonlinear Transform (MNT) from the FSA PDF and applies it to the magnitude of the input message sequence. The result is converted back to rectangular coordinates by the histo/polar_to_rec block using the unmodified phase. The vec/ramp block computes the breakpoints for plotting purposes.

Refer to the fourier/mnt block for more information on the MNT.

Inputs:

i_in	In-Phase component of input message sequence
q_in	Quadrature component of input message sequence

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram PDF

Outputs:

pdf	Plot of the FSA PDF
i_out	In-Phase component of transformed sequence
q_out	Quadrature component of transformed sequence
mnt	Plot of the FSA MNT
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
fourier/pdf, fourier/pdf_plot, fourier/mnt, histo/histo, histo/polar_to_rec, histo/rec_to_polar, vec/ramp

---

Name:
fourier/pdf

Description:
This block obtains the Fourier Series Approximation (FSA) of a Probability Density Function (PDF). This FSA method is based on a histogram of the input data magnitude.

Inputs:

bins	Histogram PDF
min	Minimum data value
max	Maximum data value

Parameters:

samples	Number of samples in data vector
P	Order of FSA

Outputs:

a	Vector of A coefficients (cosine terms)
b	Vector of B coefficients (sine terms)
T	"Period" of the FSA, equal to MAX - MIN

Name:
fourier/pdf_plot[†]

Description:
This hierarchical block plots the Fourier Series Approximation (FSA) of a Probability Density Function (PDF). A vec/minmax_ramp block generates a ramp from the minimum to the maximum input values, and the fourier/pdfplot block computes the PDF approximation for each point in the ramp.

Inputs:

a	Vector of A coefficients (cosine terms)
b	Vector of B coefficients (sine terms)
T	"Period" of the FSA, equal to MAX - MIN (T is computed by the fourier/pdf block)
message	Input message sequence

Parameters:

samples	Number of samples in data vector
points	Number of plot points for PDF
P	Order of FSA

Outputs:

pdf	FSA PDF

See also:
fourier/pdfplot, vec/minmax_ramp

Name:
fourier/pdfplot

Description:
This block plots the Fourier Series Approximation (FSA) of a Probability Density Function (PDF). The PDF approximation is computed for each point in the input, which must be a linear ramp over the desired plot range.

Inputs:

points	Input data vector
a	Vector of A coefficients (cosine terms)
b	Vector of B coefficients (sine terms)
T	"Period" of the FSA, equal to MAX - MIN (T is computed by the fourier/pdf block)

Parameters:

points	Number of plot points for PDF
P	Order of FSA

Outputs:

pdf	FSA PDF

Name:
fourier/s_nonlin_nc†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the fourier/nonlin nonlinearity. This block does not generate the summation part of the LO decision statistic.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the fourier/s_nonlin_nc block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the fourier/s_nonlin_nc block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)&

"spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence (vector)
out	Complex transformed sequence (serial)
clk_out	Timing signal: goes high when valid data sample is available at the output of fourier/s_nonlin_nc
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:

fourier/nonlin, romelib/timing, rl/inf_vsource, fourier/serial_nonlin

Name:
fourier/serial_nonlin†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the fourier/nonlin nonlinearity. In addition, the summation in the LO decision statistic is formed using the histo/correlator2 block. The output of the correlator is converted back to serial form.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the fourier/serial_nonlin block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the fourier/serial_nonlin block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence
out	Decision of nonlinearity
clk_out	Timing signal: goes high when valid data sample is available at the output of fourier/serial_nonlin
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
fourier/nonlin, histo/correlator2, romelib/timing, rl/inf_vsource

Name:
histo/ave_bp

Description:
The histogram is defined only between breakpoints, leaving values at the breakpoints undefined. However, some of the polynomial curve fitting algorithms require knowledge of the histogram values at the breakpoints. This function computes the values at the breakpoints by averaging the values of the adjacent bins. The first and last values are taken to be the values of the first and last bins. The output "histogram" will have one more data point than the input histogram.

Inputs:

in	Input histogram

Parameters:

bins	Number of bins

Outputs:

out	Output "histogram"

Name:
histo/correlator2

Description:
This block implements the matched filter - correlator for a two dimensional system. The input reference signals, I_REF and Q_REF, are assumed to have all possible signal pairs stored sequentially. The width of each signal is given by SAMPLES_PER_SYMBOL. The total number of reference signals is given by NUM_REFS.

Inputs:

i	In-Phase component of input message sequence
q	Quadrature component of input message sequence
i_ref	I-channel Reference symbols
q_ref	Q-channel Reference symbols

Parameters:

samples	Number of samples in input vectors
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference symbols

Outputs:

i_decision	Decision of In-Phase component of the message
q_decision	Decision of Quadrature component of the message

Name:

histo/enonlin†

Description:

This is a hierarchical block which applies the nonlinearity to the input signal in vector format. The histo/rec_to_polar block converts the two dimensional input signal from rectangular to polar coordinates, and the data is sorted by the vec/heap_sort block. An equiprobable bin histogram approximation of the Probability Density Function (PDF) of the magnitude is computed by the histo/equi block. The histo/mnt2 block obtains the Memoryless Nonlinear Transform (MNT) from the histogram and applies it to the magnitude of the input message sequence. The result is converted back to rectangular coordinates by the histo/polar_to_rec block using the unmodified phase.

Refer to the histo/mnt2 block for more information on the MNT.

Inputs:

i_in	In-Phase component of input message sequence
q_in	Quadrature component of input message sequence

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram PDF

Outputs:

pdf	Equiprobable bin histogram PDF
i_out	In-Phase component of transformed sequence
q_out	Quadrature component of transformed sequence
mnt	Plot of the Memoryless Nonlinear Transform
bp	Breakpoints (PDF and MNT interval boundaries)

See also:

vec/heap_sort, histo/equi, histo/mnt2, histo/rec_to_polar, histo/polar_to_rec

---

Name:
histo/equi

Description:
This block generates an equiprobable histogram Probability Density Function (PDF) and an equiprobable histogram Cumulative Distribution Function (CDF) of the input data sequence. Since The PDF and CDF are equiprobable, the probability of a data point falling in any bin is equal to 1/BINS. ***THE INPUT DATA MUST BE SORTED IN ASCENDING ORDER BEFORE IT IS APPLIED TO THIS BLOCK.***

Inputs:

data	Input data sequence (must be sorted in ascending order)

Parameters:

bins	Number of bins to generate
samples	Number of samples in data vector

Outputs:

bp	Breakpoints (PDF and CDF interval boundaries)
pdf	Equi-probable bin histogram PDF
cdf	Equi-probable bin histogram CDF

Name:
histo/histo

Description:
This block generates a histogram Probability Density Function (PDF) of the input data sequence. The width of each bin is (MAX-MIN)/BINS. The output is scaled so that the total area of the histogram is equal to 1.

Inputs:

message	Input data sequence

Parameters:

message_len	Number of samples in data vector
bins	Number of bins to generate

Outputs:

bins	Probability Density Function
min	Minimum data value
max	Maximum data value

Name:
histo/mnt2†

Description:
This block obtains the Memoryless Nonlinear Transform (MNT) from a histogram Probability Density Function (PDF). It is a hierarchical block which contains a histo/mnt_calc2 block and a histo/mnt_out2 block. The mnt_calc2 block computes the MNT, and the mnt_out2 block applies the MNT to each data point of the magnitude of the input message sequence.

This block is a generalized version of histo/mnt_iq; it has a breakpoint input so it may be used for a histogram with bins of arbitrary width. Thus, it may be used in conjunction with either a histogram or an equiprobable bin histogram. The MNT of the K^{th} histogram bin is

$$g[K] = \frac{1}{BP[K]} - \frac{\log(PDF[K+1]) - \log(PDF[K-1])}{BP[K]-BP[K-1]}$$

In the case of a one dimensional data signal, the 1/BP[K] term is not included in g[K]. For a two dimensional data signal, this block operates only on the magnitude.

The G_PLOT output is the MNT computed for each histogram bin. Thus, the length of the G_PLOT vector must be the same as the BINS vector. The length of the G_VAL vector is the same as the DATA vector.

Inputs:

bins	Histogram PDF
bp	Breakpoints (PDF and MNT interval boundaries)
data	Input message sequence (1 or 2 dimensional)

Parameters:

bins	Number of bins in the PDF
samples	Number of samples in data vector
dim	Dimensionality of the message sequence (1 or 2)

Outputs:

g_val	MNT of each data sample
g_plot	Plot of MNT from minimum to maximum data value

See also:
histo/mnt_calc2, histo/mnt_out2

Name:
histo/mnt_calc2

Description:
This function computes the Memoryless Nonlinear Transform (MNT)for a histogram with bins of arbitrary width. The MNT of the K^{th} histogram bin is

$$g[K] = \frac{1}{BP[K]} - \frac{\log(PDF[K+1]) - \log(PDF[K-1])}{BP[K] - BP[K-1]}$$

In the case of a one dimensional data signal, the 1/BP[K] term is not included in g[K]. For a two dimensional data signal, this block operates only on the magnitude.

Inputs:

bins	Histogram Probability Density Function (PDF)
bp	Breakpoints (PDF and MNT interval boundaries)

Parameters:

bins	Number of bins in the PDF
dim	Dimensionality of the message sequence (1 or 2)

Outputs:

g	MNT for each histogram bin

Name:
histo/mnt_calc_2d

Description:
This function computes the Memoryless Nonlinear Transform (MNT) for a histogram with bins of equal width. The MNT of the K^{th} histogram bin is

$$g[K] = \frac{1}{R} - \frac{\log(PDF[K+1]) - \log(PDF[K-1])}{WIDTH}$$

where WIDTH = MAX - MIN and
R = MIN + K * WIDTH.

This block operates on the magnitude of the two dimensional data signal.

Inputs:

bins	Histogram Probability Density Function (PDF)
min	Minimum magnitude of input message sequence
max	Maximum magnitude of input message sequence

Parameters:

bins	Number of bins in the PDF

Outputs:

g	MNT for each histogram bin

Name:
histo/mnt_iq†

Description:
This block obtains the Memoryless Nonlinear Transform (MNT) from a histogram Probability Density Function (PDF). It is a hierarchical block which contains a histo/mnt_calc_d2 block and a histo/mnt_out block. The mnt_calc_2d block computes the MNT, and the mnt_out block applies the MNT to each data point of the magnitude of the input message sequence.

This block assumes that each histogram bin has equal width. In addition, the input message sequence is assumed to be the magnitude of a two dimensional signal. The MNT of the K^{th} histogram bin is

$$g[K] = \frac{1}{R} - \frac{\log(PDF[K+1]) - \log(PDF[K-1])}{WIDTH}$$

where WIDTH = MAX - MIN and
R = MIN + K * WIDTH.

The G_PLOT output is the MNT computed for each histogram bin. Thus, the length of the G_PLOT vector must be the same as the BINS vector. The length of the G_VAL vector is the same as the MESSAGE vector.

Inputs:

bins	Histogram PDF
min	Minimum magnitude of input message sequence
max	Maximum magnitude of input message sequence
message	Input message sequence (2 dimensional)

Parameters:

bins	Number of bins in the PDF
samples	Number of samples in data vector

Outputs:

g_val	MNT of each data sample
g_plot	Plot of MNT from minimum to maximum data value

See also:
histo/mnt_calc_2d, histo/mnt_out

Name:
histo/mnt_out

Description:
This block applies the Memoryless Nonlinear Transform (MNT) generated by the histo/mnt_calc_2d block to each sample in the input message sequence. This block assumes that each histogram bin has equal width. In addition, the input message sequence is assumed to be the magnitude of a two dimensional signal.

Inputs:

g	MNT for input message sequence
min	Minimum magnitude of input message sequence
max	Maximum magnitude of input message sequence
message	Input message sequence (2 dimensional)

Parameters:

bins	Number of bins in the Probability Density Function
samples	Number of samples in data vector

Outputs:

g_val	MNT of each input sample

Name:
histo/mnt_out2

Description:
This block applies the Memoryless Nonlinear Transform (MNT) generated by the histo/mnt_calc2 block to each sample in the input message sequence.This block permits the use of arbitrarily spaced breakpoints.

Inputs:

g	MNT for input message sequence
bp	Breakpoints (MNT interval boundaries)
data	Input message sequence (1 or 2 dimensional)

Parameters:

bins	Number of bins in the histogram
samples	Number of samples in data vector

Outputs:

g_val	MNT of each data sample

Name:
histo/nonlin6†

Description:

This is a hierarchical block which applies the nonlinearity to the input signal in vector format. The histo/rec_to_polar block converts the two dimensional input signal from rectangular to polar coordinates, and a histogram approximation of the Probability Density Function (PDF) of the magnitude is computed by the histo/histo block. The histo/mnt_iq block obtains the Memoryless Nonlinear Transform (MNT) from the histogram and applies it to the magnitude of the input message sequence. The result is converted back to rectangular coordinates by the histo/polar_to_rec block using the unmodified phase. The vec/ramp block computes the breakpoints for plotting purposes.

Refer to the histo/mnt_iq block for more information on the MNT.

Inputs:

i_in	In-Phase component of input message sequence
q_in	Quadrature component of input message sequence

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram PDF

Outputs:

pdf	Histogram PDF
i_out	In-Phase component of transformed sequence
q_out	Quadrature component of transformed sequence
mnt	Plot of the Memoryless Nonlinear Transform
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/histo, histo/mnt_iq, histo/rec_to_polar, histo/polar_to_rec, vec/ramp

Name:
histo/polar_to_rec

Description:
This block accepts a complex vector in polar coordinates and converts it to a complex vector in rectangular coordinates as follows:

I=R*cos(A)
Q=R*sin(A)

where R is the magnitude and A is the phase.

Inputs:

mag	Magnitude of input data
phase	Phase of input data

Parameters:

samples	Number of samples in input and output vectors

Outputs:

i	In-Phase component of input data
q	Quadrature component of input data

Name:
histo/prob_error†

Description:
This hierarchical block computes the Probability of Bit Error (Pb) of a serial data stream. The number of errors in both the I and Q channel are summed over SYMBOLS_PER_CALC data bits in each channel. The Pb is equal to this sum divided by (SYMBOLS_PER_CALC*2).

The CLOCK_IN control signal must go high when a sample is present at the input. This signal may be generated using the CLK_OUT output of the romelib/timing block.

Inputs:

i_in	Reference signal--I channel
i_dec	Decision of receiver--I channel
q_dec	Decision of receiver--Q channel
q_in	Reference signal--Q channel
clock_in	Clock enable

Parameters:

symbols_per_calc	Number of symbols per Pb calculation
samples_per_symbol	Number of samples per data symbol

Outputs:

pe	Probability of Bit Error
pe_clock	Clock signal for Pb signal sink

Name:
histo/rec_to_polar

Description:
This block accepts a complex vector in rectangular coordinates and converts it to a complex vector in polar coordinates as follows:

$$R=sqrt(I^2+Q^2)$$
$$A=atan(Q/I)$$

where R is the magnitude and A is the phase.

In the implementation of this block, the phase ranges from -pi/2 to 3*pi/2, instead of the usual -pi to pi. This makes it possible to avoid using an extra IF statement in the program. The two forms are mathematically equivalent.

Inputs:

i	In-Phase component of input data
q	Quadrature component of input data

Parameters:

samples	Number of samples in input and output vectors

Outputs:

mag	Magnitude of input data
phase	Phase of input data

Name:

histo/s_enonlin[†]

Description:

This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the histo/enonlin nonlinearity. In addition, the summation in the LO decision statistic is formed using the histo/correlator2 block. The output of the correlator is converted back to serial form.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the histo/s_enonlin block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the histo/s_enonlin block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence
out	Decision of nonlinearity
clk_out	Timing signal: goes high when valid data sample is available at the output of histo/s_enonlin
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/enonlin, histo/correlator2, romelib/timing, rl/inf_vsource

---

Name:
histo/s_enonlin_nc†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the histo/enonlin nonlinearity. This block does not generate the summation part of the LO decision statistic.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the histo/s_enonlin_nc block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the histo/s_enonlin_nc block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence (vector)
out	Complex transformed sequence (serial)
clk_out	Timing signal: goes high when valid data sample is available at the output of histo/s_enonlin_nc
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/enonlin, romelib/timing, rl/inf_vsource, histo/s_enonlin

Name:
histo/s_nonlin_nc†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the histo/nonlin6 nonlinearity. This block does not generate the summation part of the LO decision statistic.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the histo/s_nonlin_nc block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the histo/s_nonlin_nc block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence (vector)
out	Complex transformed sequence (serial)
clk_out	Timing signal: goes high when valid data sample is available at the output of histo/s_nonlin_nc
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/nonlin6, romelib/timing, rl/inf_vsource, histo/serial_nonlin

---

Name:
histo/serial_nonlin†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the histo/nonlin6 nonlinearity. In addition, the summation in the LO decision statistic is formed using the histo/correlator2 block. The output of the correlator is converted back to serial form.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the histo/serial_nonlin block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the histo/serial_nonlin block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr

("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence
out	Decision of nonlinearity
clk_out	Timing signal: goes high when valid data sample is available at the output of histo/serial_nonlin
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/nonlin6, histo/correlator2, romelib/timing, rl/inf_vsource

Name:
jam/channel†

Description:
This block adds Gaussian noise and jammers to the transmitted data sequence. Currently only three Continuous Wave (CW) jammers and a Partial Band (PB) are implemented. Any other interference blocks may be easily added. (All power parameters are measured in dB)

Inputs:

in	Transmitted Message

Parameters:

J1_S	Ratio of First CW jammer power to signal power
phase1	First CW phase
freq1	First CW frequency
J2_S	Ratio of Second CW jammer power to signal power
phase2	Second CW phase
freq2	Second CW frequency
J3_S	Ratio of Third CW jammer power to signal power
phase3	Third CW phase
freq3	Third CW frequency
Jpb_S	Ratio of PB jammer power to signal power
filt_order	Lowpass filter order
atten	Passband ripple in dB
pfreq	Passband 3 dB edge frequency
sfreq	Stopband 3 dB edge frequency
Eb_No	Ratio of Bit Energy to Gaussian noise power
No	Gaussian noise power
s_freq	Sampling frequency
samples_per_symbol	Number of Samples per Information Symbol

Outputs:

out	Corrupted Message

See also:
rl/cw_jammer, rl/pb_jammer

Name:
linear/serial_lin†

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the histo/correlator2 block. The output of the correlator is converted back to serial form.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the linear/serial_lin block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the linear/serial_lin block. and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

out	Decision of linear receiver
clk_out	Timing signal: goes high when valid data sample is available at the output of linear/serial_lin
hold_vec	Timing signal: goes low when vector outputs are available

See also:

histo/correlator2, romelib/timing, rl/inf_vsource

Name:
mipa/nonlin†

Description:
This is a hierarchical block which applies the nonlinearity to the input signal in vector format. The histo/rec_to_polar block converts the two dimensional input signal from rectangular to polar coordinates, and a MIPA of the Probability Density Function (PDF) of the magnitude is computed by the mipa/pdf block. The poly/mnt block obtains the Memoryless Nonlinear Transform (MNT) from the MIPA PDF and applies it to the magnitude of the input message sequence. The result is converted back to rectangular coordinates by the histo/polar_to_rec block using the unmodified phase. The poly/plot block generates a plot of the PDF from the polynomial coefficients.

Refer to the poly/mnt block for more information on the MNT.

Inputs:

i_in	In-Phase component of input message sequence
q_in	Quadrature component of input message sequence

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the histogram PDF
points	Number of plot points for PDF and MNT

Outputs:

pdf	CPA PDF
i_out	In-Phase component of transformed sequence
q_out	Quadrature component of transformed sequence
mnt	Plot of the Memoryless Nonlinear Transform
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
histo/rec_to_polar, mipa/pdf, poly/mnt, poly/plot, histo/polar_to_rec

Name:
mipa/pdf

Description:
This block obtains the M-Interval Polynomial Approximation (MIPA) of a Probability Density Function (PDF). The MIPA is a concatenation of polynomial curves, which minimizes the squared error between the approximation and the actual PDF. This implementation only supports 0th, 2nd, and 4th order MIPAs.

Inputs:

message	Input message sequence

Parameters:

bins	Number of bins
samples	Number of samples in input message sequence
order	Order of MIPA

Outputs:

bp	Breakpoints (PDF interval boundaries)
coeff	Polynomial coefficients values a_0, a_1, .., a_P

Name:
mipa/s_nonlin_nc[†]

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the mipa/nonlin nonlinearity. This block does not generate the summation part of the LO decision statistic.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the mipa/s_nonlin_nc block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the mipa/s_nonlin_nc block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence (vector)
out	Complex transformed sequence (serial)
clk_out	Timing signal: goes high when valid data sample is available at the output of mipa/s_nonlin_nc
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
mipa/nonlin, romelib/timing, rl/inf_vsource, mipa/serial_nonlin

Name:
mipa/serial_nonlin[†]

Description:
This hierarchical block buffers the received data and converts it from serial to vector data and applies it to the mipa/nonlin nonlinearity. In addition, the summation in the LO decision statistic is formed using the histo/correlator2 block. The output of the correlator is converted back to serial form.

The romelib/timing block generates timing waveforms which enable and disable the nonlinearity and all subsequent blocks at appropriate times. Of particular interest are the CLK_OUT and HOLD_VEC signals. The HOLD_VEC signal holds vector blocks while the serial data is buffered into vector form. This signal goes low during the simulation iteration in which the vector processing completes and the outputs are available. The CLK_OUT signal provides flexibility for use in multirate systems by pulsing high when a valid data sample is produced at the output of the mipa/serial_nonlin block. The CLK_OUT signal has the same period as the CLK_IN signal, thus output samples are produced at the same rate as that of the input signal. The remaining romelib/timing signals are internal to the mipa/serial_nonlin block, and are used in serial to vector and vector to serial buffering.

The SAMPLES_PER_SYMBOL, NUM_REFS, and M_TYPE are characteristics of the reference symbols. The reference signals must be created before using this block, and are stored in the /spwdata/ref library. This permits the designer to use any desired signal constellation. The NUM_REFS parameter must be exported to the histo/correlator2 block, and the other two parameters are exported to the filename of the rl/inf_vsource block as "ref/"&substr ("m_type":model,2,length("m_type":model)-2)&"_"&xstring("samples_per_symbol":model)& "spb_i'" for the I channel signals and "ref/"&substr("m_type":model,2,length("m_type": model)-2) &"_"&xstring("samples_per_symbol":model)&"spb_q'" for the Q channel signals. For example, the filenames for a Binary Phase Shift Keying (BPSK) signal with 25 samples per symbol would be /spwdata/ref/bpsk_25spb_i and /spwdata/ref/bpsk_25spb_q.

Inputs:

in	Input message sequence (complex input)
clk_in	Clock input for romelib/timing

Parameters:

samples	Number of samples in input vectors
bins	Number of bins in the Probability Density Function (PDF)
samples_per_symbol	Number of samples per data symbol
num_refs	Number of reference signals
m_type	Modulation type

Outputs:

mnt	Plot of the Memoryless Nonlinear Transform
g	Complex transformed sequence
out	Decision of nonlinearity
clk_out	Timing signal: goes high when valid data sample is available at the output of histo/serial_nonlin
hold_vec	Timing signal: goes low when vector outputs are available
pdf	Plot of the PDF of magnitude
bp	Breakpoints (PDF and MNT interval boundaries)

See also:
mipa/nonlin, histo/correlator2, romelib/timing, rl/inf_vsource

Name:
poly/mnt†

Description:
This block obtains the Memoryless Nonlinear Transform (MNT) from a polynomial approximation to a Probability Density Function (PDF). It is a hierarchical block which contains two poly/mnt_calc blocks. One of them computes the MNT for each value of the magnitude R of the input message sequence, and the other generates a plot of the MNT from the minimum to the maximum R value with the aid of a vec/minmax_ramp block. This block only operates on the magnitude of the input message sequence. For a P^{th} order polynomial the MNT is

$$g[R] = \frac{1}{R} - \frac{a_1 + 2a_2R + \cdots + Pa_PR^{P-1}}{a_0 + a_1R + a_2R^2 + \cdots + a_PR^P}$$

If the LT parameter is set to 1, the linear transform will be applied to the coefficients. That means that the R values in the K^{th} bin must range from 0 to the width of the K^{th} bin (BP[K]-BP[K-1]), whereas without the transform the R values in each bin will range from BP[K-1] to BP[K].

Inputs:

bp	Breakpoints (PDF and MNT interval boundaries)
coeff	PDF polynomial coefficients values a_0, a_1, .., a_P
data	Input message sequence (1 or 2 dimensional)

Parameters:

samples	Number of samples used to compute MNT
bins	Number of bins
points	Number of plot points in G_PLOT
order	Polynomial order (number of coefficients - 1)
lt	Linear transform flag

Outputs:

g_val	MNT of input data
g_plot	Plot of MNT from minimum to maximum data value

See also:
poly/mnt_calc, vec/minmax_ramp

Name:
poly/mnt_calc

Description:
This function computes the Memoryless Nonlinear Transform (MNT)from a polynomial approximation to a Probability Density Function (PDF) for each value of the magnitude R of the input message sequence. For a P^{th} order polynomial the MNT is

$$g[R] = \frac{1}{R} - \frac{a_1 + 2a_2R + \cdots + Pa_pR^{P-1}}{a_0 + a_1R + a_2R^2 + \cdots + a_pR^P}$$

In the case of a one dimensional data signal, the 1/R term is not included in g[R]. For a two dimensional data signal, this block operates only on the magnitude R.

If the LT parameter is set to 1, the linear transform will be applied to the coefficients. That means that the R values in the K^{th} bin must range from 0 to the width of the K^{th} bin (BP[K]-BP[K-1]), whereas without the transform the R values in each bin will range from BP[K-1] to BP[K].

Inputs:

bp	Breakpoints (PDF and MNT interval boundaries)
coeff	PDF polynomial coefficients values a_0, a_1, .., a_P
data	Input message sequence (1 or 2 dimensional)

Parameters:

bins	Number of bins
samples	Number of samples used to compute MNT
order	Polynomial order (number of coefficients - 1)
dim	Dimensions of data signal (1 or 2)
lt	Linear transform flag

Outputs:

mnt	MNT for polynomial

Name:
poly/mnt_calc_gt

Description:
This function computes the Memoryless Nonlinear Transform (MNT)from a polynomial approximation to a Probability Density Function (PDF) for each value of the magnitude R of the input message sequence. For a P^{th} order polynomial the MNT is

$$g[R] = \frac{1}{R} - \frac{a_1 + 2a_2R + \cdots + Pa_PR^{P-1}}{a_0 + a_1R + a_2R^2 + \cdots + a_PR^P}$$

In the case of a one dimensional data signal, the 1/R term is not included in g[R]. For a two dimensional data signal, this block operates only on the magnitude R.

The first and last bins are fitted with the MNT of a Gaussian PDF instead of a polynomial MNT. ***THIS BLOCK IS TO BE USED IN CONJUNCTION WITH THE CPA/PDF_GT BLOCK.***

If the LT parameter is set to 1, the linear transform will be applied to the coefficients. That means that the R values in the K^{th} bin must range from 0 to the width of the K^{th} bin (BP[K]-BP[K-1]), whereas without the transform the R values in each bin will range from BP[K-1] to BP[K].

Inputs:

bp	Breakpoints (PDF and MNT interval boundaries)
coeff	PDF polynomial coefficients values a_0, a_1, .., a_P
data	Input message sequence (1 or 2 dimensional)

Parameters:

bins	Number of bins
samples	Number of samples used to compute MNT
order	Polynomial order (number of coefficients - 1)
dim	Dimensions of data signal (1 or 2)
lt	Linear transform flag

Outputs:

mnt	MNT for polynomial

Name:
poly/mnt_gt†

Description:
This block obtains the Memoryless Nonlinear Transform (MNT) from a polynomial approximation to a Probability Density Function (PDF). It is a hierarchical block which contains two poly/mnt_calc_gt blocks. One of them computes the MNT for each value of the magnitude R of the input message sequence, and the other generates a plot of the MNT from the minimum to the maximum R value with the aid of a vec/minmax_ramp block. This block only operates on the magnitude of the input message sequence. For a P^{th} order polynomial the MNT is

$$g[R] = \frac{1}{R} - \frac{a_1 + 2a_2R + \cdots + Pa_PR^{P-1}}{a_0 + a_1R + a_2R^2 + \cdots + a_PR^P}$$

The first and last bins are fitted with the MNT of a Gaussian PDF instead of a polynomial MNT. ***THIS BLOCK IS TO BE USED IN CONJUNCTION WITH THE CPA/PDF_GT BLOCK.***

If the LT parameter is set to 1, the linear transform will be applied to the coefficients. That means that the R values in the K^{th} bin must range from 0 to the width of the K^{th} bin (BP[K]-BP[K-1]), whereas without the transform the R values in each bin will range from BP[K-1] to BP[K].

Inputs:

bp	Breakpoints (PDF and MNT interval boundaries)
coeff	PDF polynomial coefficients values a_0, a_1, .., a_P
data	Input message sequence (1 or 2 dimensional)

Parameters:

bins	Number of bins
samples	Number of samples used to compute MNT
order	Polynomial order (number of coefficients - 1)
lt	Linear transform flag

Outputs:

g_val	MNT of input data
g_plot	Plot of MNT from minimum to maximum data value

See also:
poly/mnt_calc_gt, vec/minmax_ramp

Name:
poly/plot

Description:
This function plots a piecewise polynomial curve with the number of intervals equal to BINS. The polynomial in each interval is of the form

$$y[X] = a_0 + a_1 X + a_2 X^2 + \dots + a_P x^P$$

where P is the ORDER. The X value is varied from its minimum to maximum value to generate the plot. The number of points in this plot is equal to POINTS.

If the LT parameter is set to 1, the linear transform will be applied to the coefficients. That means that the X values in the K^{th} bin must range from 0 to (BP[K]-BP[K-1]), whereas without the transform the X values in each bin will range from BP[K-1] to BP[K].

Inputs:

bp	X-axis Breakpoints (interval boundaries)
coeff	Polynomial coefficients values a_0, a_1, .., a_P

Parameters:

bins	Number of bins
points	Number of points to plot
order	Polynomial order (number of coefficients - 1)
lt	Linear transform flag

Outputs:

plot	Polynomial curve

Name:
poly/plot_gt

Description:
This function plots a piecewise polynomial curve with the number of intervals equal to BINS. The polynomial in each interval is of the form

$$y[X] = a_0 + a_1 X + a_2 X^2 + \dots + a_P x^P$$

where P is the ORDER. The X value is varied from its minimum to maximum value to generate the plot. The number of points in this plot is equal to POINTS.

If the LT parameter is set to 1, the linear transform will be applied to the coefficients. That means that the X values in the K^{th} bin must range from 0 to (BP[K]-BP[K-1]), whereas without the transform the X values in each bin will range from BP[K-1] to BP[K].

The first and last bins are fitted with a Gaussian PDF instead of a polynomial.

Inputs:

bp	X-axis Breakpoints (interval boundaries)
coeff	Polynomial coefficients values a_0, a_1, .., a_P

Parameters:

bins	Number of bins
points	Number of points to plot
order	Polynomial order (number of coefficients - 1)
lt	Linear transform flag

Outputs:

plot	Polynomial curve

Name:
rl/complex_cw†

Description:
This block generates a single Continuous Wave (CW) complex sinusoid with a specified amplitude, frequency, and phase.

Inputs:
none

Parameters:

amp	CW amplitude
phase	CW phase
freq	CW frequency
s_freq	Sampling frequency

Outputs:

cw_out	Jammer output

Name:
rl/complex_data†

Description:
This block generates a random Quadrature Phase Shift Keying (QPSK) data sequence. A fundamental relationship exists between the symbol rate and the sampling frequency, namely S_FREQ = Rs * SAMPLES_PER_SYMBOL. This means that all three of these parameters may not vary independently, but one must be a function of the other two. For this reason, the symbol rate parameter is exported as Rs = S_FREQ / SAMPLES_PER_SYMBOL.

Inputs:
none

Parameters:

amp	Bit amplitude
Rs	Symbol rate
prob_zero	Probability of a bit being zero
s_freq	Sampling frequency

Outputs:

data_out	QPSK data stream

Name:
rl/cw_jammer†

Description:
This block generates three Continuous Wave (CW) jammers with specified amplitude, frequency, and phase. Any number of CW jammers up to three may be obtained by setting the amplitude of the undesired jammers to an arbitrarily small value.

Inputs:
none

Parameters:

amp1	First CW amplitude
phase1	First CW phase
freq1	First CW frequency
amp2	Second CW amplitude
phase2	Second CW phase
freq2	Second CW frequency
amp3	Third CW amplitude
phase3	Third CW phase
freq3	Third CW frequency
s_freq	Sampling frequency

Outputs:

cw_out	Jammer output

See also:
rl/complex_cw

Name:
rl/hold†

Description:
This block generates an output that goes high every SAMPLES iterations. There is no pulse on the 0th iteration.

Inputs:
none

Parameters:

samples	Number of iterations between pulses

Outputs:

hold	Output clock

Name:
rl/inf_vsource†

Description:
This block is identical to the spb/vsource SPW library block except that when an end of file (EOF) occurs the last output is retained for the remainder of the simulation. This block detects the EOF from the spb/vsource and holds the spb/vsource from then on.

Inputs:
none

Parameters:
same as spb/vsource

Outputs:
same as spb/vsource

See also:
spb/vsource

Name:
rl/pb_jammer†

Description:
This block generates a baseband Partial Band (PB) jammer by low pass filtering Gaussian noise with an elliptic lowpass filter. The Passband and Stopband edge frequencies are assumed to be normalized by the symbol rate.

Inputs:
none

Parameters:

Jpb_S	Ratio of jammer power to signal power in dB
filt_order	Lowpass filter order
atten	Passband ripple in dB
pfreq	Passband 3 dB edge frequency
sfreq	Stopband 3 dB edge frequency
s_freq	Sampling frequency

Outputs:

pb_out	Jammer output

Name:
rl/psk_err_cnt†

Description:
THIS BLOCK DOES NOT WORK CORRECTLY BECAUSE COMM/REAL_ERR_CNT DOES NOT WORK CORRECTLY.

Inputs:

X	Actual transmitted message
Y	Decision of receiver

Parameters:

Outputs:

pe	Probability of error
num_symbols	
results_clk	

See also:
rl/real_err_cnt

Name:
rl/real_err_cnt[†]

Description:
THIS BLOCK DOES NOT WORK CORRECTLY BECAUSE COMM/REAL_ERR_CNT DOES NOT WORK CORRECTLY.

Inputs:

X	Actual transmitted message
Y	Decision of receiver

Parameters:

Outputs:

pe	Probability of error
num_symbols	
results_clk	

See also:
rl/real_err_cnt

Name:
romelib/timing†

Description:
This block generates the timing signals for the nonlinear receiver and all subsequent blocks. In particular, it allows for serial-to-vector conversion of the input data to the nonlinearity blocks, and vector-to-serial conversion at the output. The CLK_IN input is provided to facilitate the integration of the nonlinear blocks into multirate systems. This input should be tied to a timing source which goes high on every simulation iteration during which a valid input data sample is present. The only parameter for this block, SAMPLES, should be set to the length of the data vector which will be processed by the nonlinear block.

The sequence of output signal states is as follows:

a) The HOLD_IN output goes low every iteration that the CLK_IN input goes high and indicates when a valid input signal sample is present.
b) The LOADOUT output goes high during the simulation iteration when the last of the SAMPLES samples is available at the input. This is a signal to the circular_buffer block that all the required data samples are present and that the data vector is complete.
c) The HOLD_VEC output goes low one simulation iteration after the last of the SAMPLES samples is available at the input. This signal should be tied to the HOLD input of the nonlinearity blocks as well as all vector output sinks/blocks associated with the nonlinearity block, allowing these blocks to go active during the simulation iteration when all the required data samples are present in vector form.
d) The LOADIN goes high one simulation iteration after the last of the SAMPLES samples is available at the input. This signal is used to "latch" the output signal vector of the nonlinearity block for vector-to-serial conversion.
e) The HOLD_OUT signal goes low one simulation iteration after each iteration when the CLK_IN input goes high. This signal is provided for clocking the serial blocks located after the nonlinearity block.
f) The CLK_OUT output first goes high on the same simulation iteration as the HOLD_VEC output goes low and the LOADIN output goes high, and then it periodically goes high after this with a period equal to that of the CLK_IN input signal. The CLK_OUT signal is used to indicate when a valid serial data sample is present at the output of the nonlinearity block.

Inputs:

clk_in	Synchronizing clock input

Parameters:

samples	Length of input data vector

Outputs:

hold_in	Activation signal for serial blocks preceding the nonlinearity block
loadout	Indicates when input data vector has been filled
hold_vec	Activation signal for the vector processing/output blocks
loadin	"Latches" the output data vector for vector-to-serial conversion
hold_out	Activation signal for serial blocks following the nonlinearity block
clk_out	Synchronizing clock output

Name:
vec/heap_sort

Description:
This block heap sorts the input data in ascending order. This sort is of order N log (N). For a description of the heap sort algorithm see Numerical Recipes in C (1988).

Inputs:

in	Input vector

Parameters:

points	Number of points in the vector

Outputs:

out	Sorted vector

Name:
vec/minmax

Description:
This block finds the minimum and maximum data points in the given data sequence.

Inputs:

in	Input data vector

Parameters:

length	Number of points in input vector

Outputs:

min	Minimum value
max	Maximum value

Name:
vec/stretch_nl

Description:
This block plots a histogram by copying the value of each bin to the output a number of times that is proportional to the width of the bin. The total width of the histogram is T_WIDTH = BP[BINS] - BP[0], and the width of the K^{th} bin is WIDTH[K] = BP[K] - BP[K-1]. The K^{th} bin value is copied to the output N times, where N = POINTS * WIDTH[K] / T_WIDTH.

The number of breakpoints is one more than the number of input points.

Inputs:

bp	Breakpoints of histogram
in	Input histogram

Parameters:

bins	Number of bins in the input sequence
points	Number of points in the output sequence

Outputs:

out	Plot of histogram

Name:
vec/minmax_ramp†

Description:
This block generates a linear ramp of length RAMP_LEN from the minimum to the maximum values in the given data sequence. If RAMP_LEN = 1 it is not possible to generate a line through both min and max unless min = max. In the event that RAMP_LEN = 1, RAMP is set equal to the average of min and max.

Inputs:

in	Input data vector

Parameters:

data_len	Input data vector length
ramp_len	Ramp vector length

Outputs:

ramp	Output ramp

See also:
vec/minmax, vec/ramp

Name:
vec/ramp

Description:
This block generates a linear ramp of length POINTS from MIN to MAX. If POINTS = 1 it is not possible to generate a line through both MIN and MAX unless MIN = MAX. In the event that POINTS = 1, the ramp is equal to the average of MIN and MAX.

Inputs:

min	Minimum data value
max	Maximum data value

Parameters:

points	Ramp vector length

Outputs:

ramp	Output ramp

Name:
vec/var

Description:
This block computes the mean and variance of the input data vector as follows:

$$MEAN = \frac{1}{LEN} \sum_{i=0}^{LEN-1} IN[i] \qquad VAR = \frac{1}{LEN-1} \sum_{i=0}^{LEN-1} (IN[i] - MEAN)^2$$

Inputs:

in	Input data vector

Parameters:

len	Input vector length

Outputs:

mean	Mean of the input data sequence
var	Variance of the input data sequence

Appendix B
SPW Block Diagrams

This Appendix is an alphabetical listing of the details of the hierarchical IIT SPW blocks that were used to generate the results in Volume 1 of this report.

When the BDE prints a block diagram, the parameters do not reflect their exported values. Rather, the default values for these parameters are displayed. However, in the actual simulation the parameter values have all been correctly exported.

CPA/CDF2

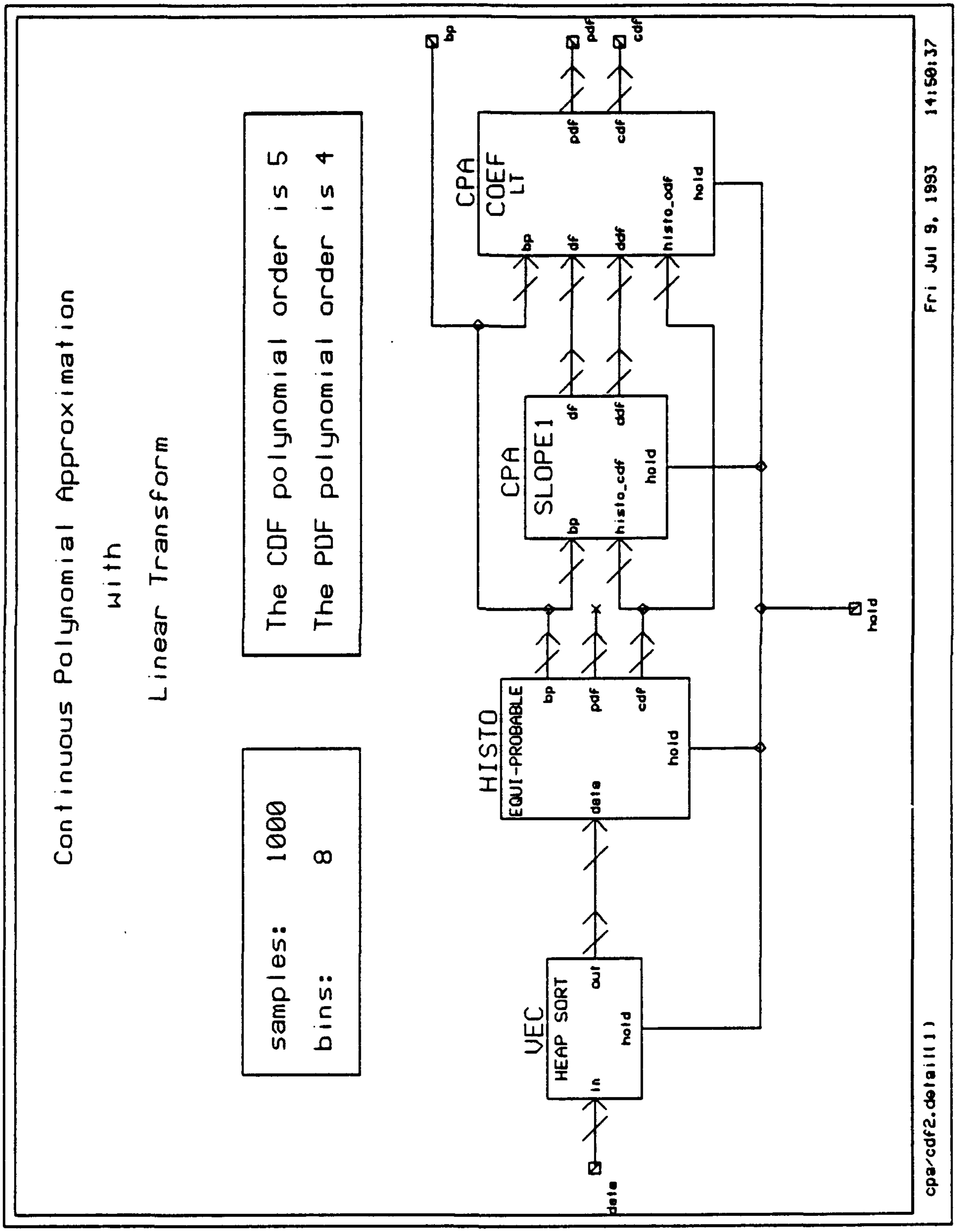


Figure (B-1)

CPA/CDF2_NONLIN

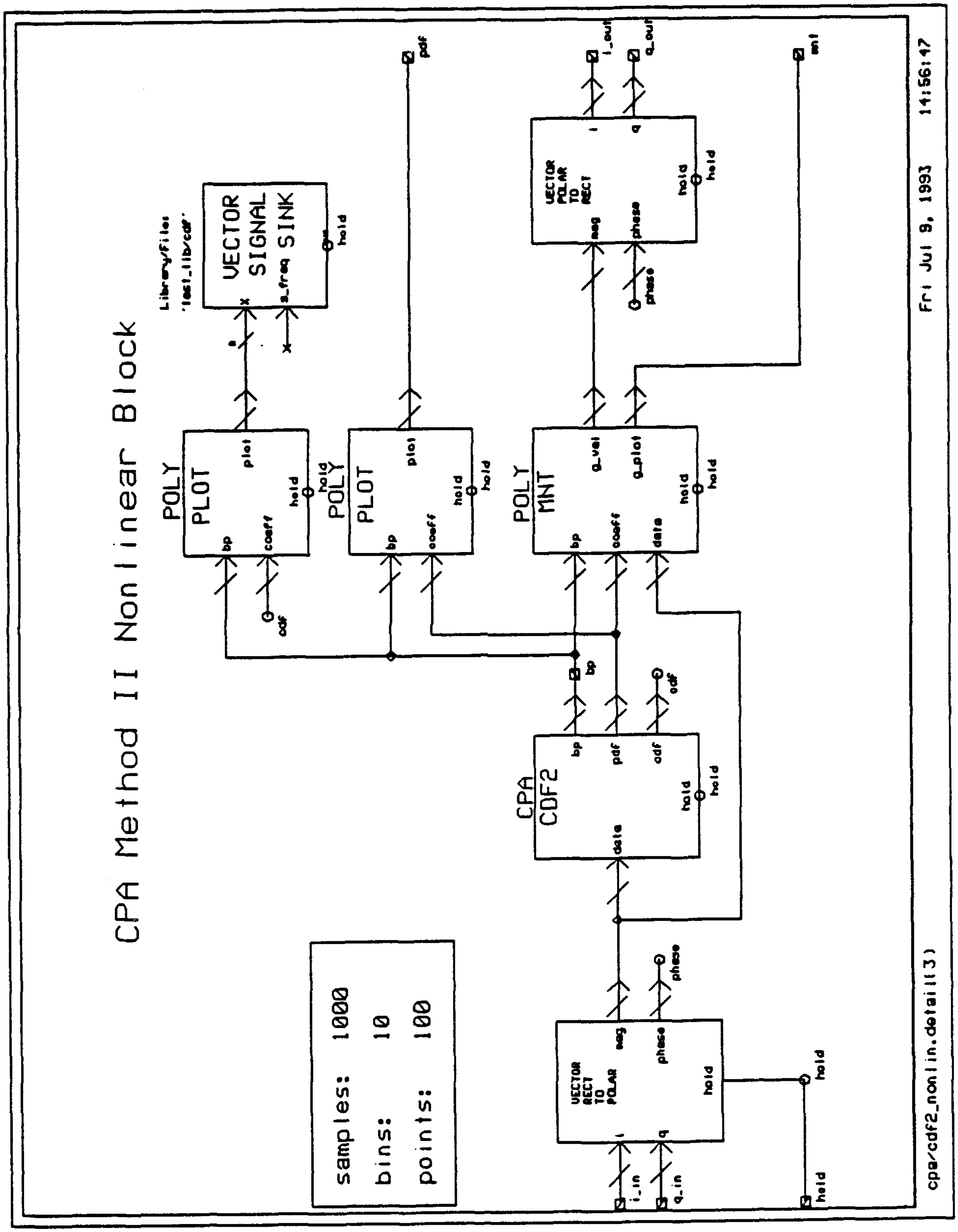


Figure (B-2)

CPA/CDF3

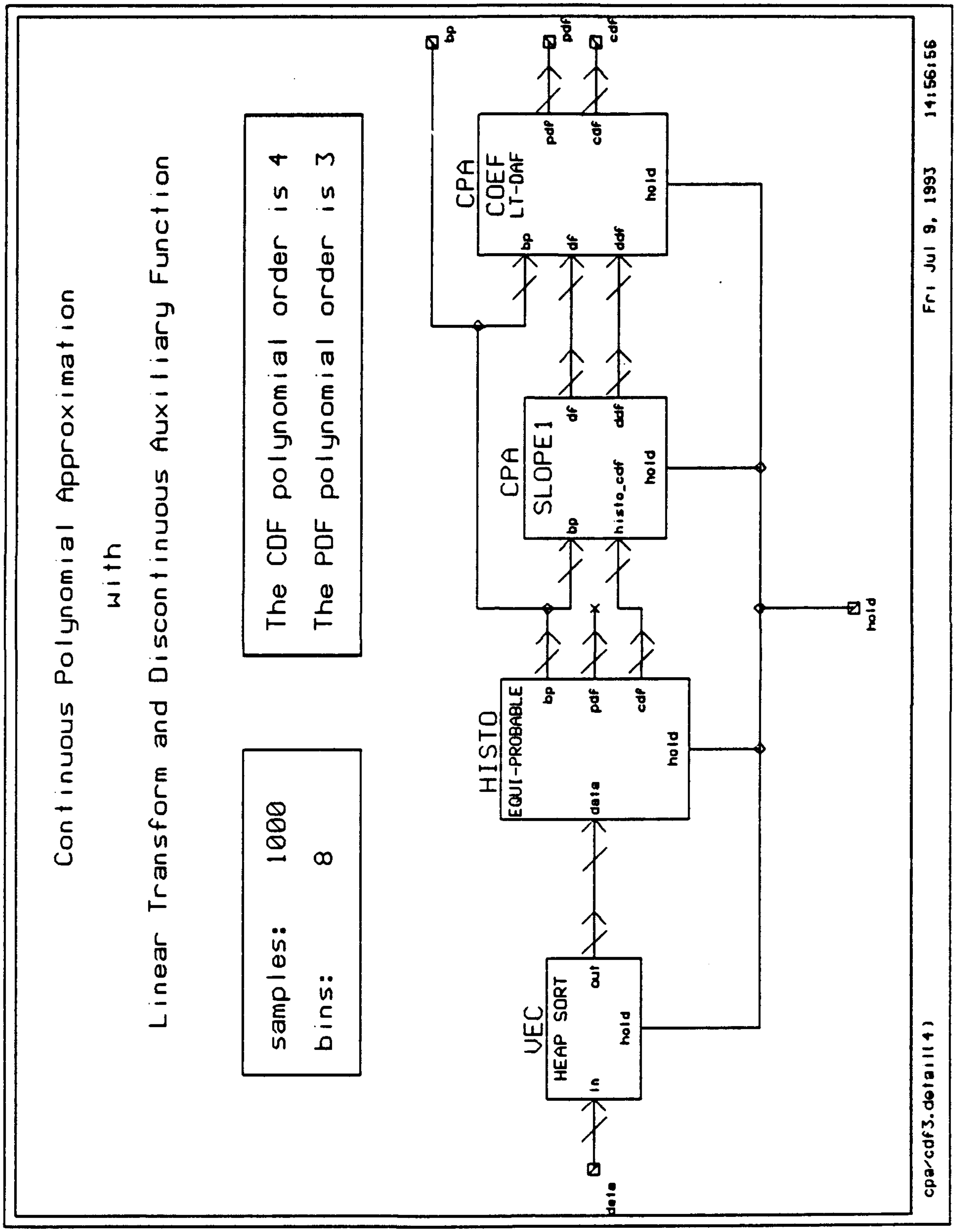


Figure (B-3)

CPA/CDF3_NONLIN

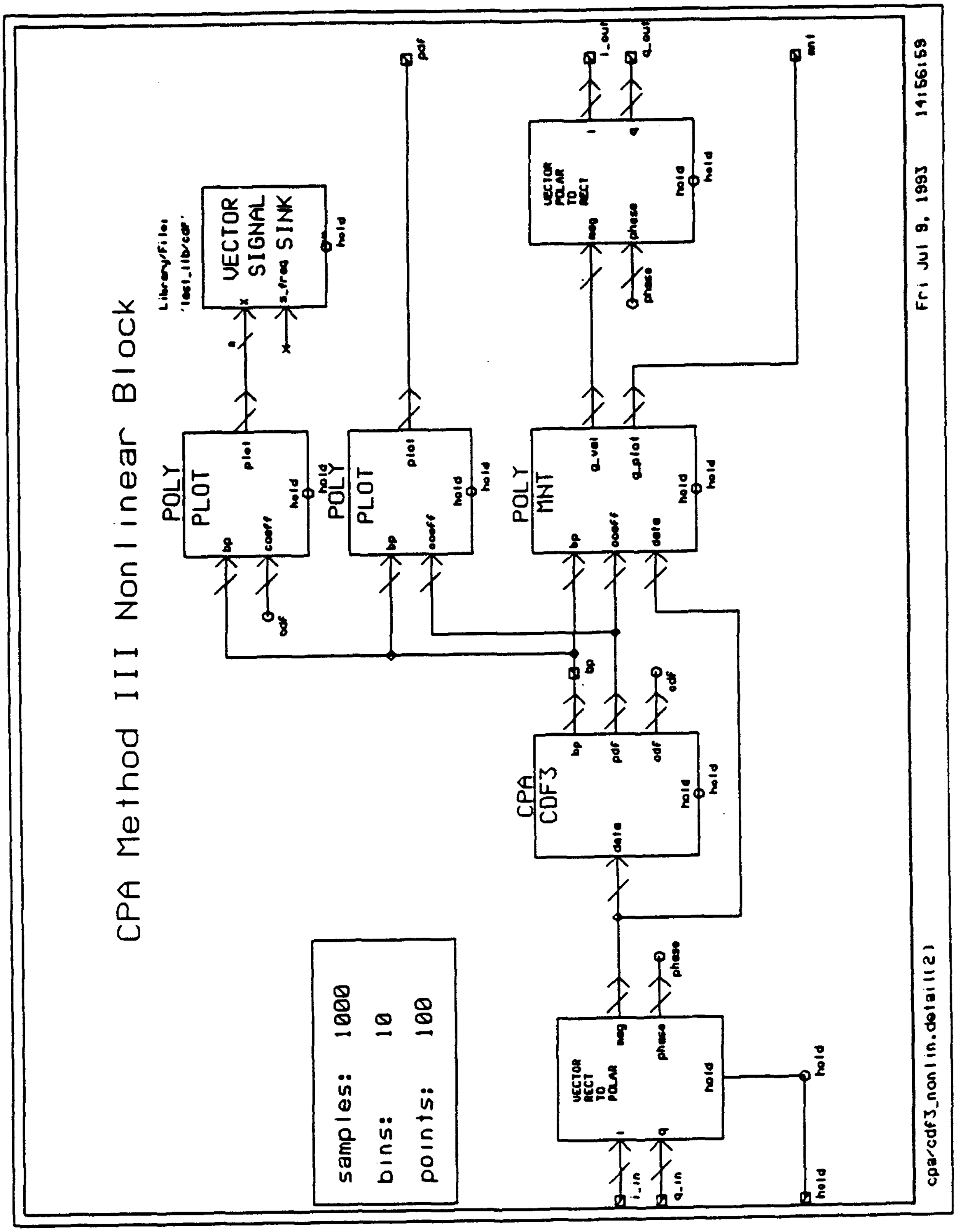


Figure (B-4)

CPA/PDF_GT

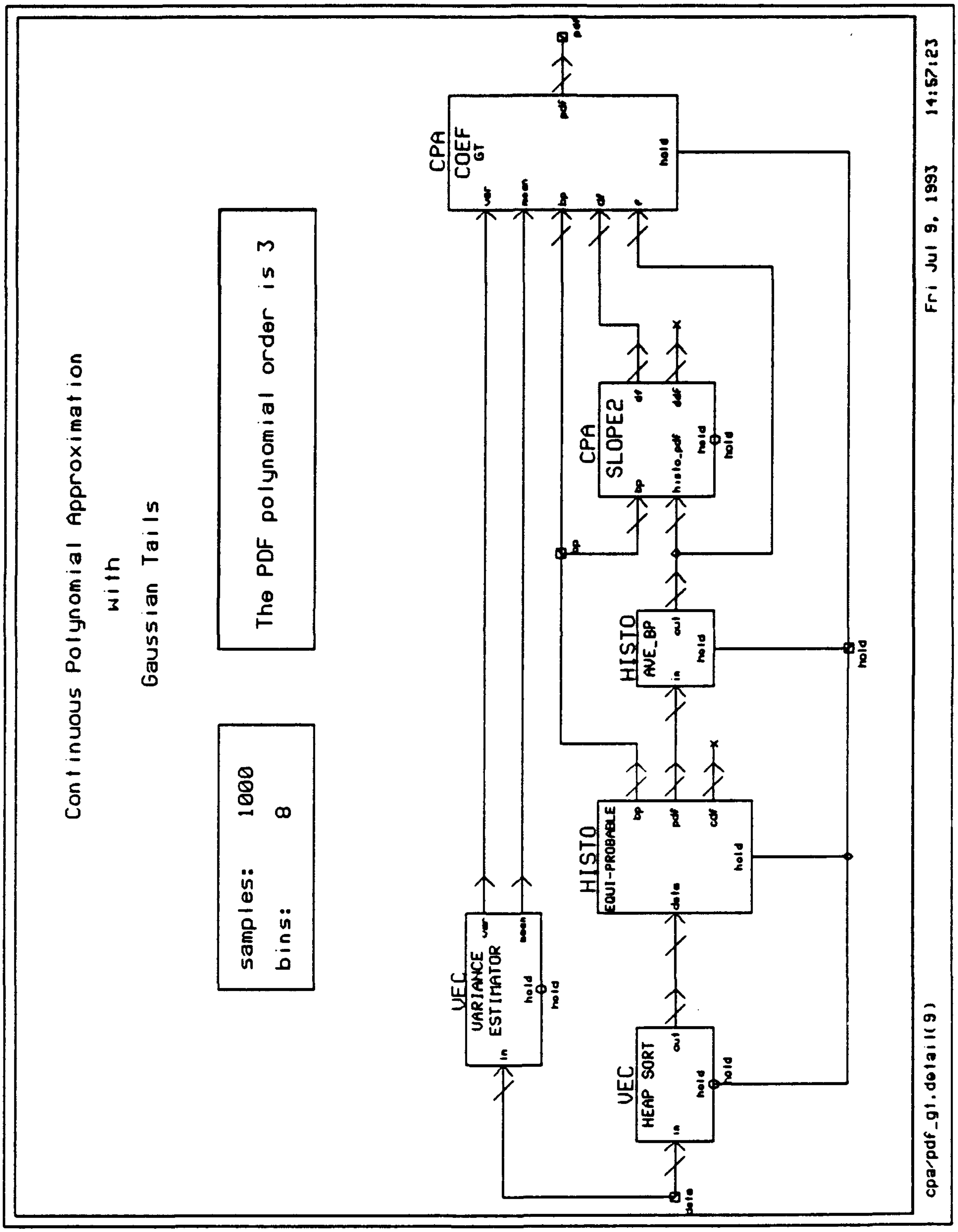


Figure (B-5)

CPA/PDF_GT_NONLIN

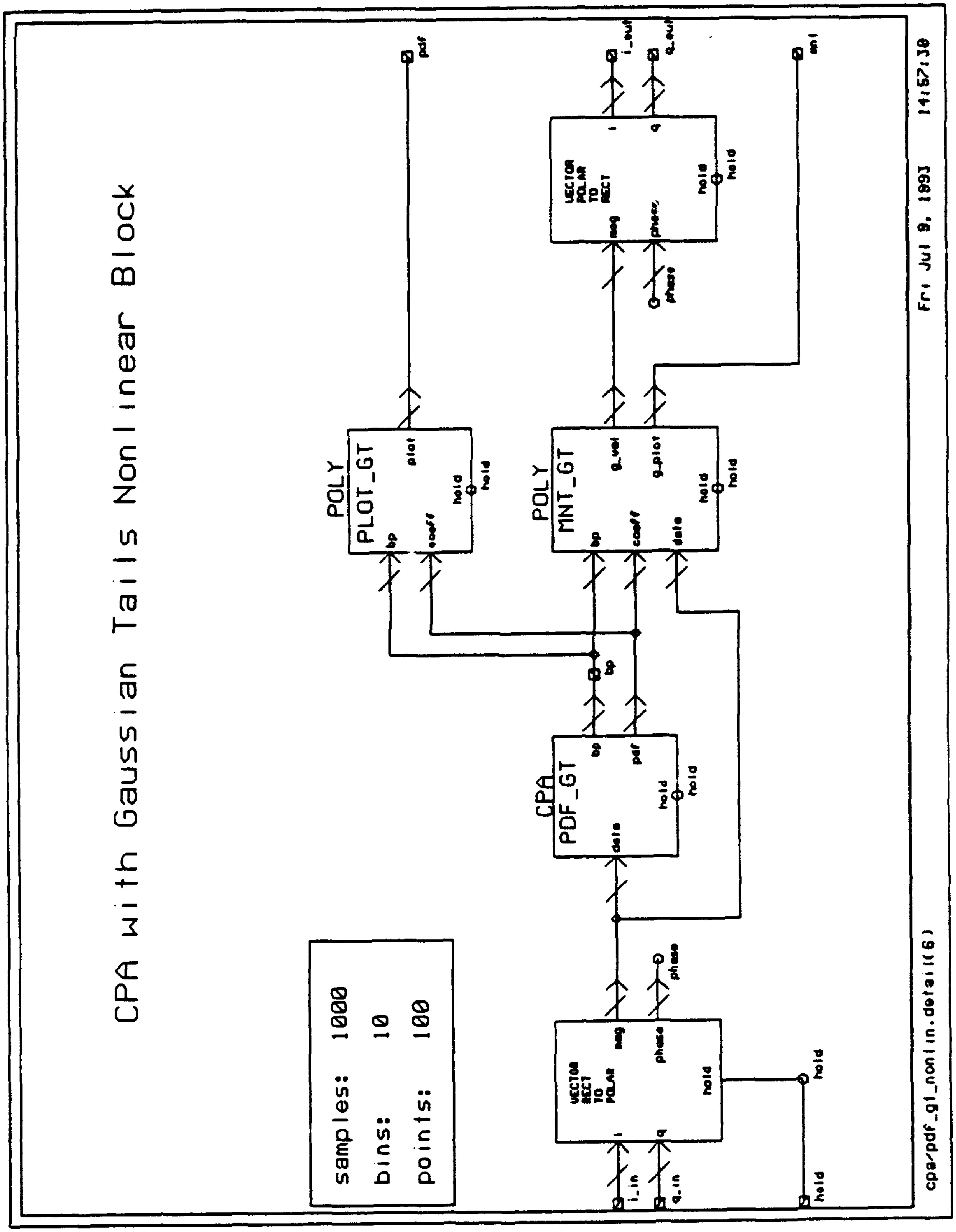


Figure (B-6)

CPA/S_CDF2_NL_NC

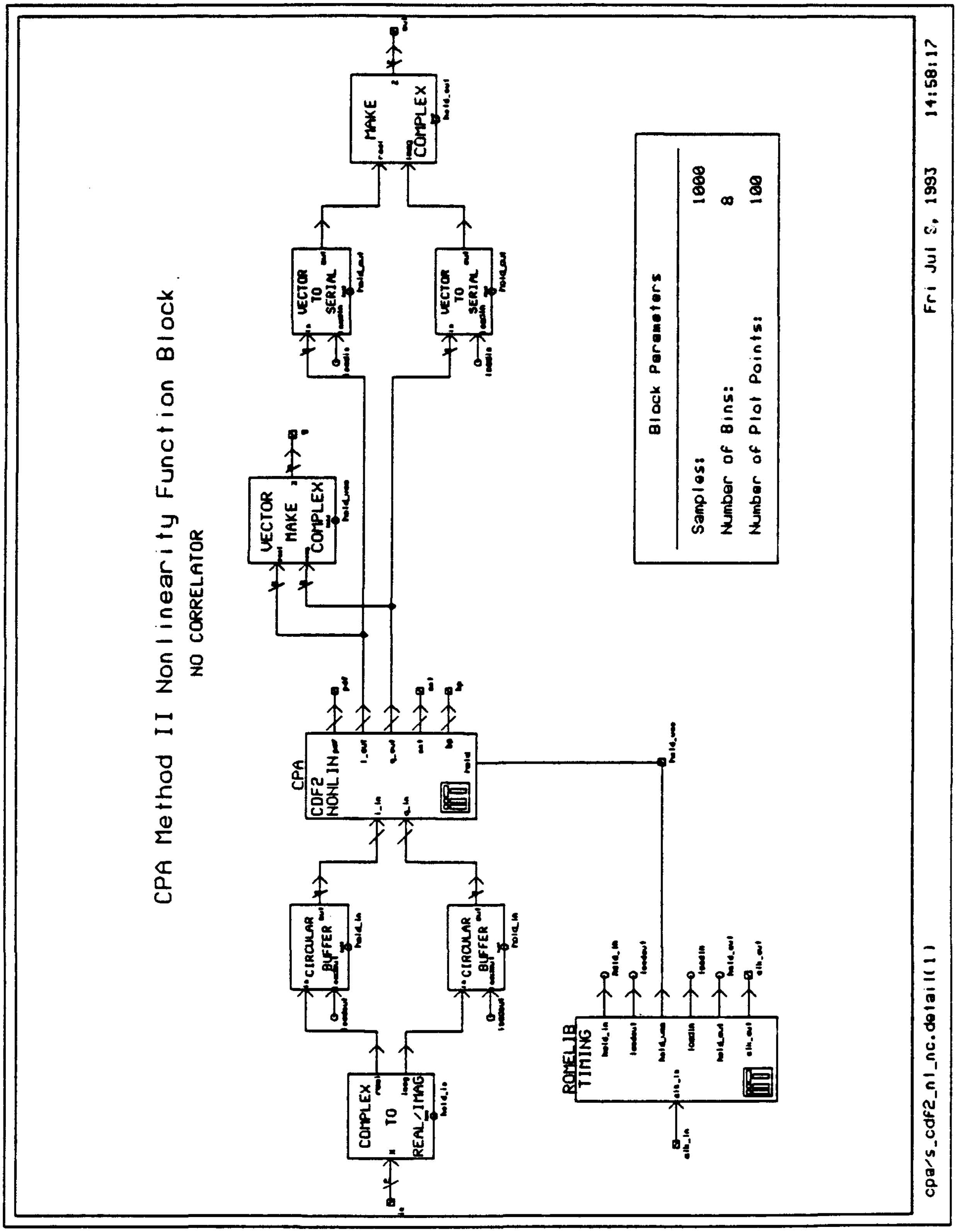


Figure (B-7)

CPA/S_CDF2_NONLIN

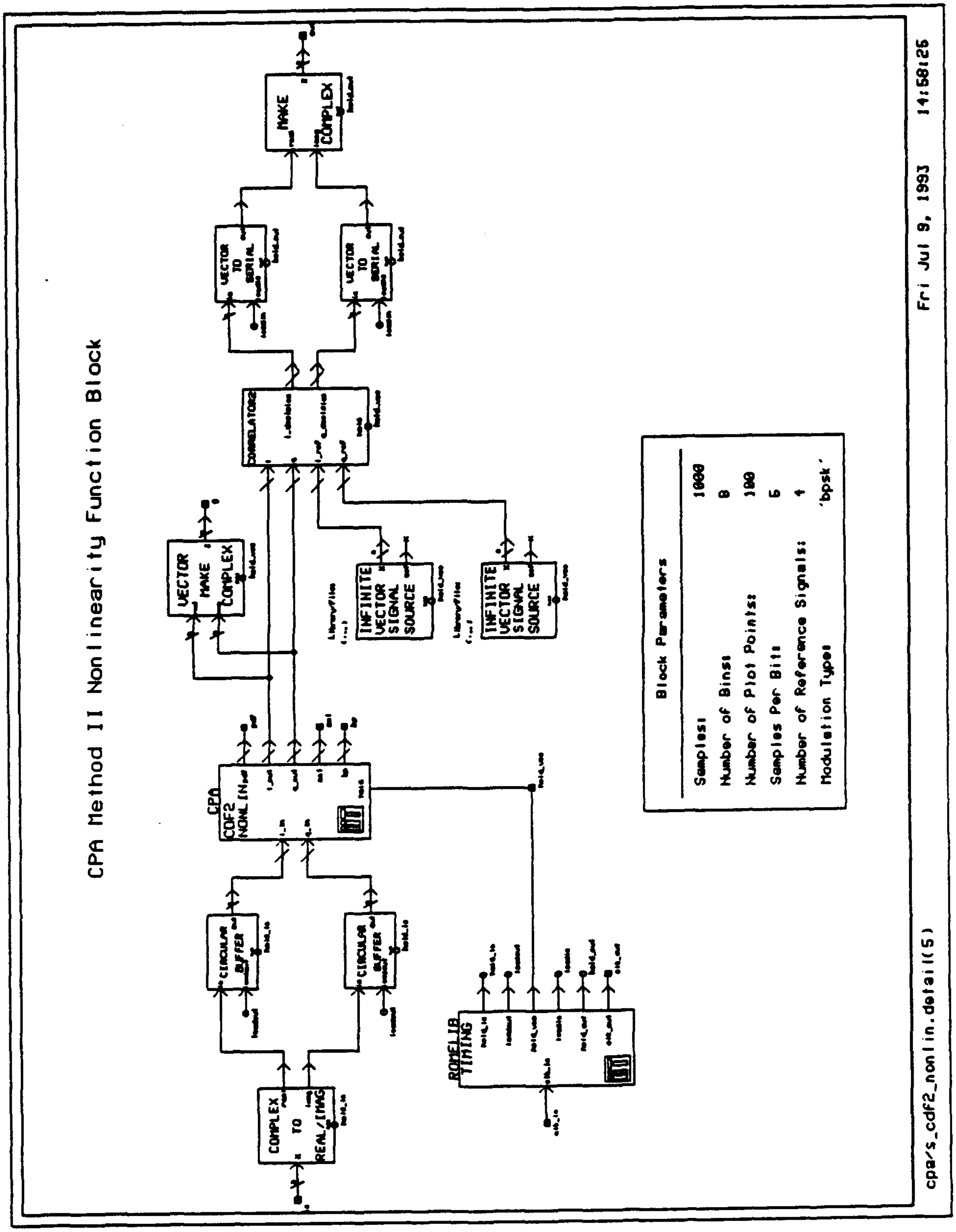


Figure (B-8)

CPA/S_CDF3_NL_NC

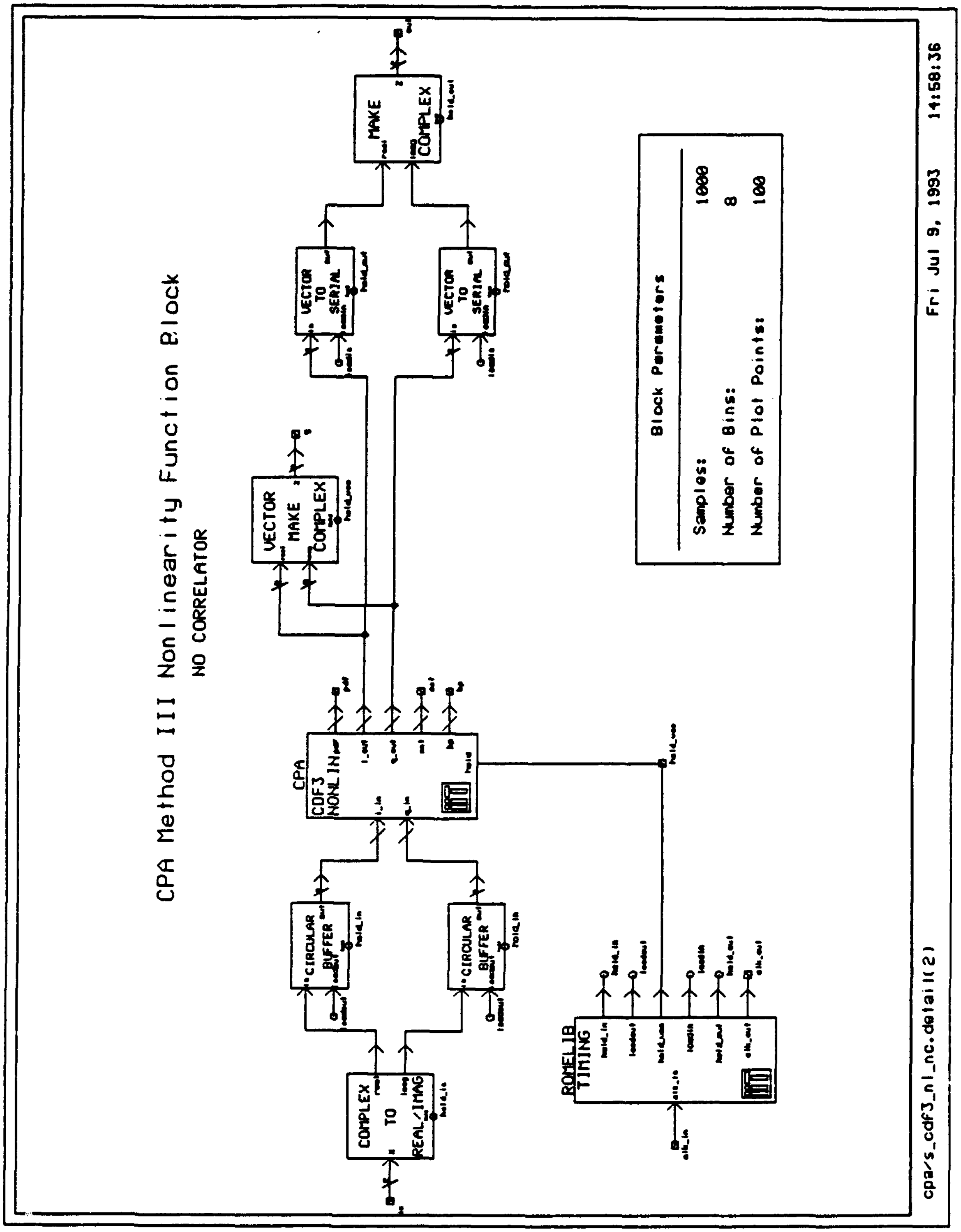


Figure (B-9)

CPA/S_CDF3_NONLIN

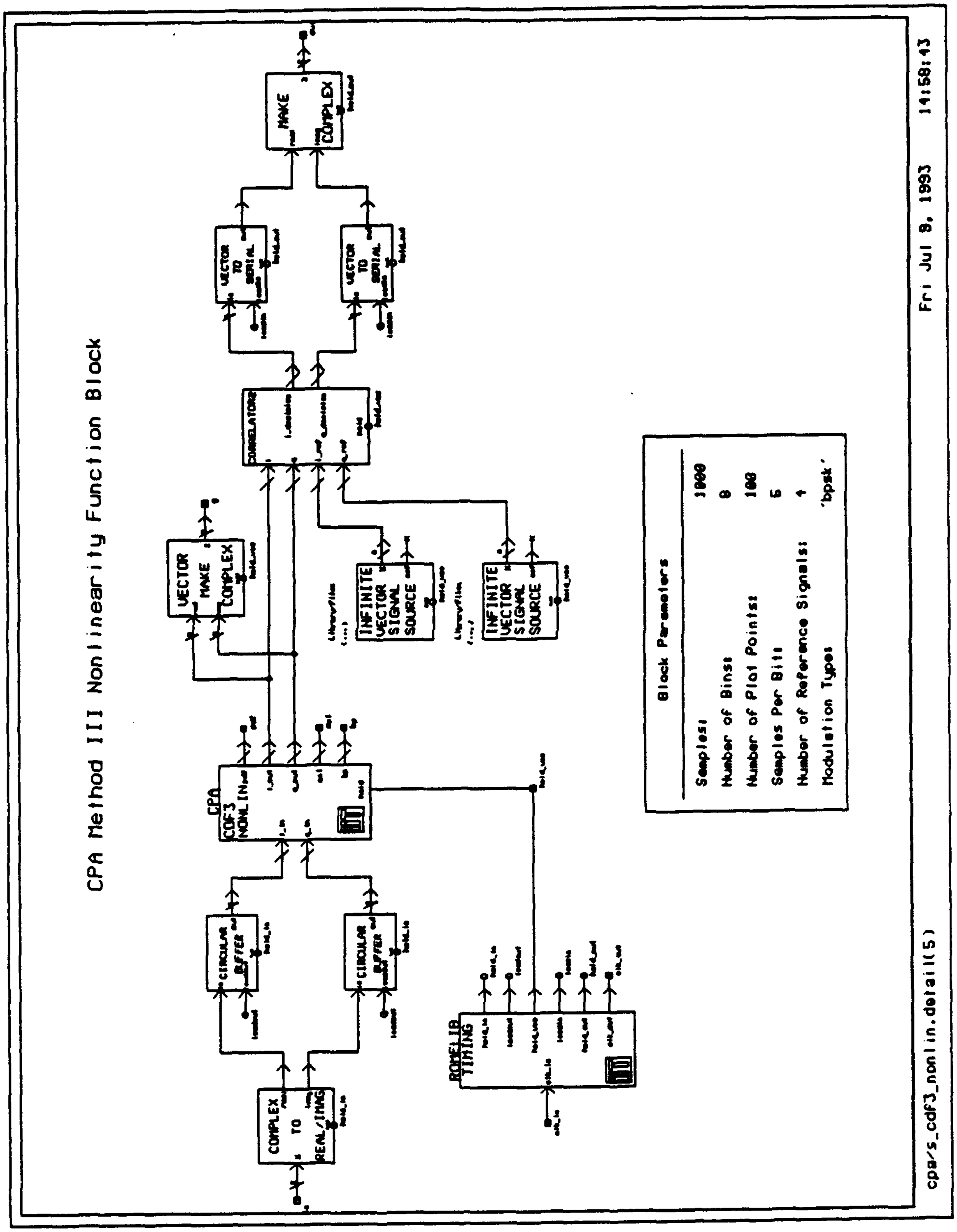


Figure (B-10)

CPA/S_PDF_GT_NL_NC

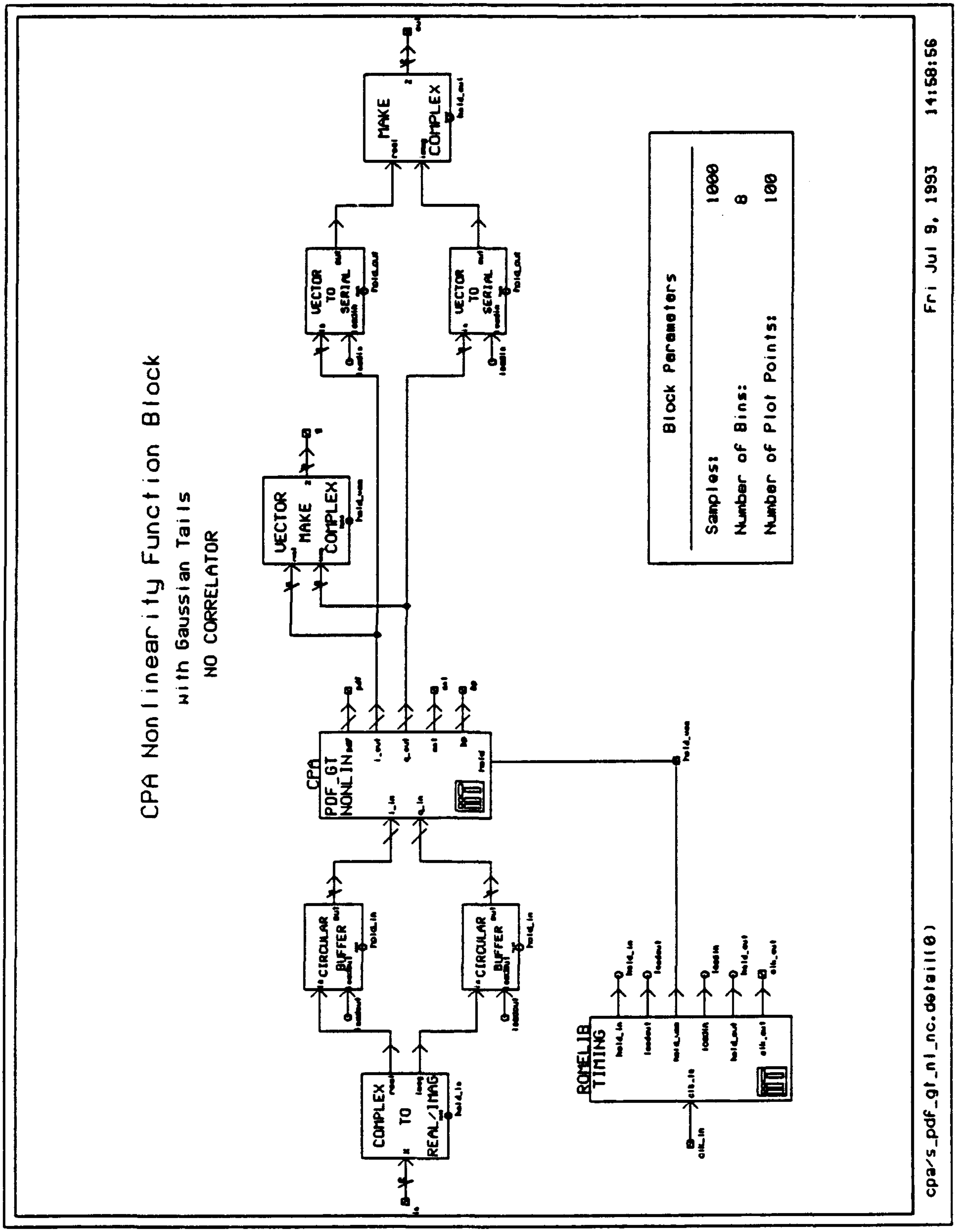


Figure (B-11)

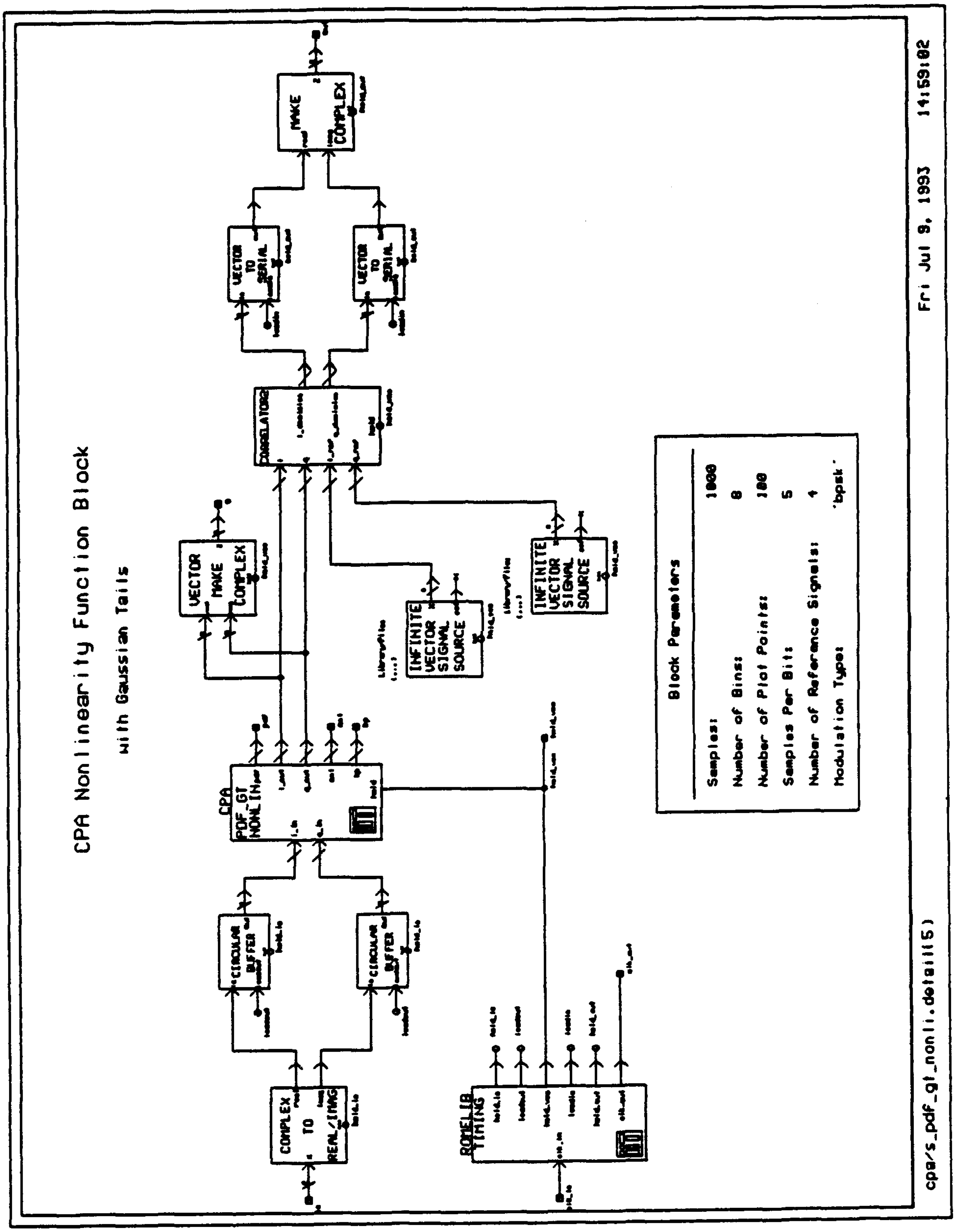

CPA Nonlinearity Function Block
with Gaussian Tails
COMPLEX TO REAL/IMAG
CIRCULAR BUFFER
CIRCULAR BUFFER
CPA PDF_GT NONLIN
VECTOR MAKE COMPLEX
INFINITE VECTOR SIGNAL SOURCE
INFINITE VECTOR SIGNAL SOURCE
CORRELATOR2
VECTOR TO SERIAL
VECTOR TO SERIAL
MAKE COMPLEX
ROMELIB TIMING
Block Parameters
Samples: 1000
Number of Bins: 8
Number of Plot Points: 100
Samples Per Bit: 5
Number of Reference Signals: 4
Modulation Type: 'bpsk'
cpa/s_pdf_gt_nonli.detail(5)
Fri Jul 9, 1993 14:59:02

Figure (B-12)

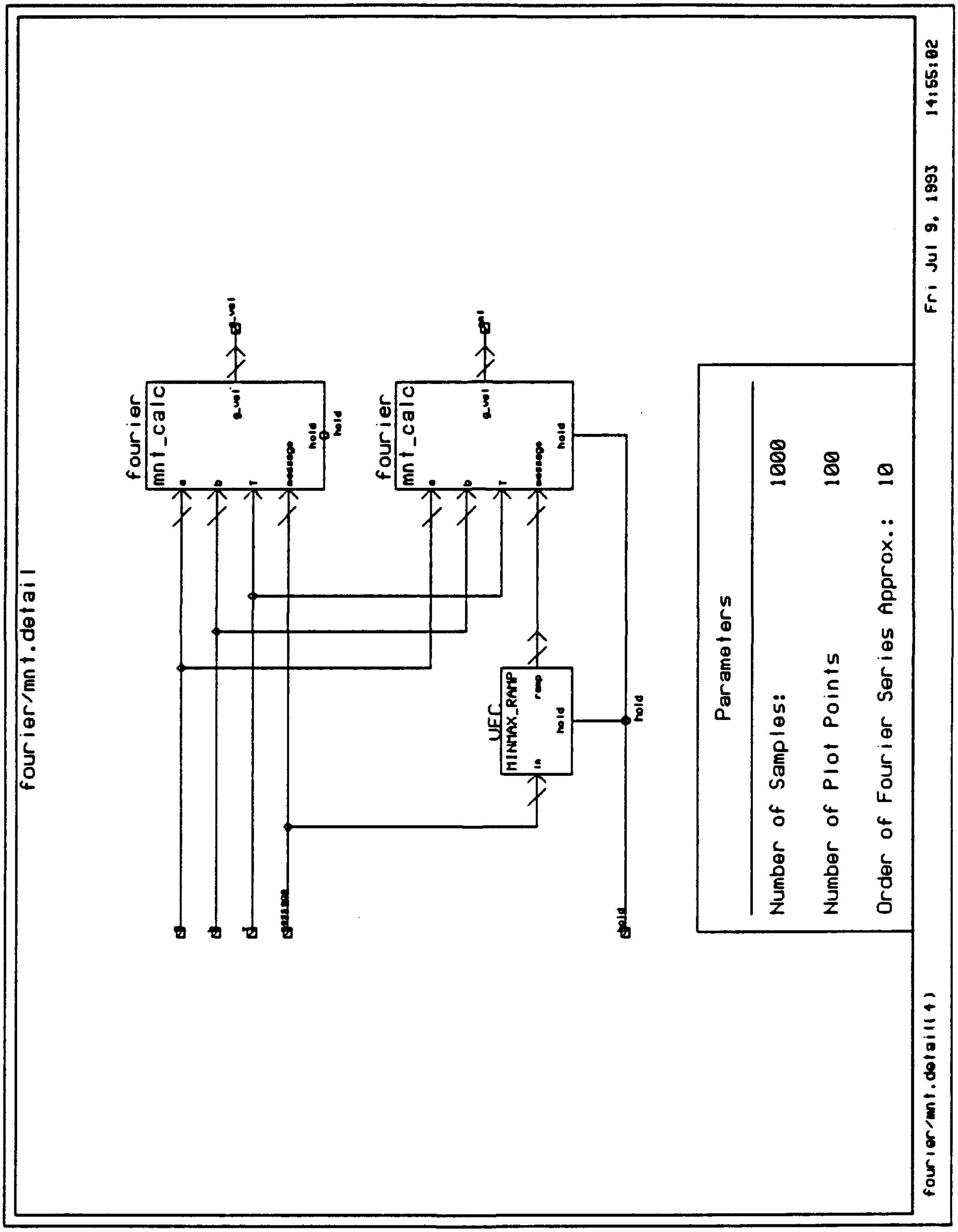

fourier/mnt.detail
fourier
mnt_calc
g_vel
hold
message
fourier
mnt_calc
g_vel
message
hold
UFC
MINMAX_RAMP
ramp
hold
in
hold
Parameters
Number of Samples: 1000
Number of Plot Points 100
Order of Fourier Series Approx.: 10
Fri Jul 9, 1993 14:55:02
fourier/mnt.detail(4)

Figure (B-13)

FOURIER/NONLIN

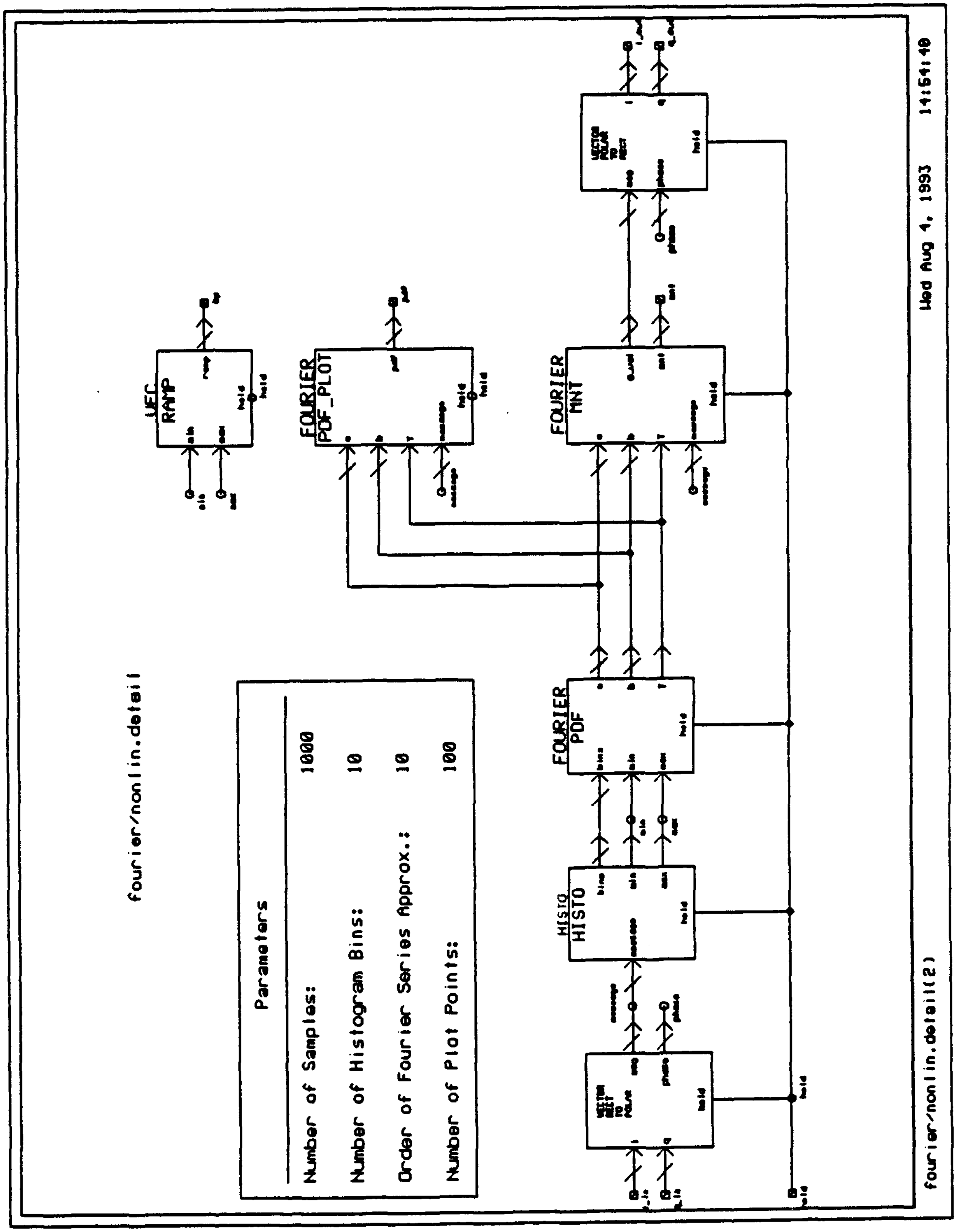


Figure (B-14)

FOURIER/PDF_PLOT

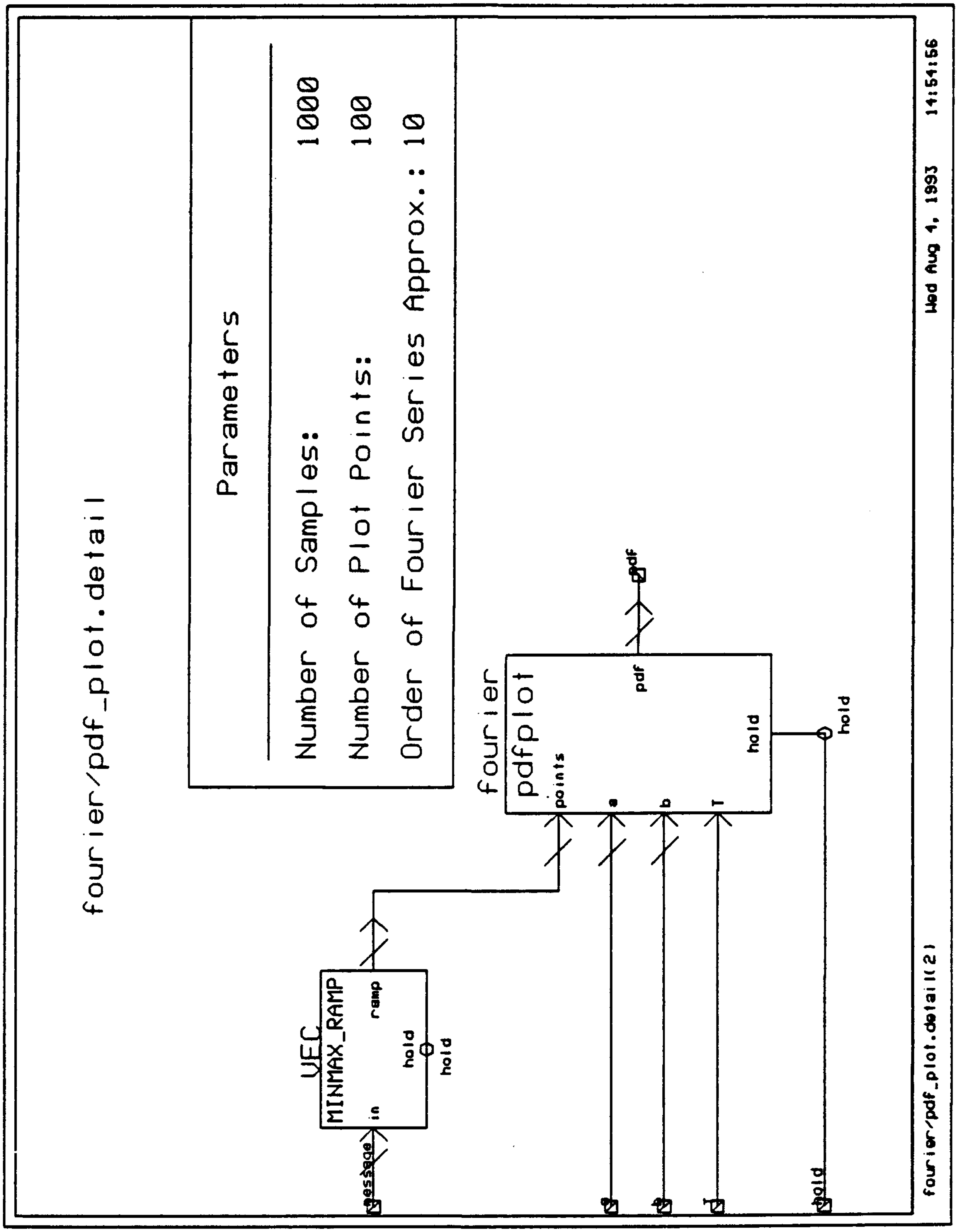


Figure (B-15)

FOURIER/S_NONLIN_NC

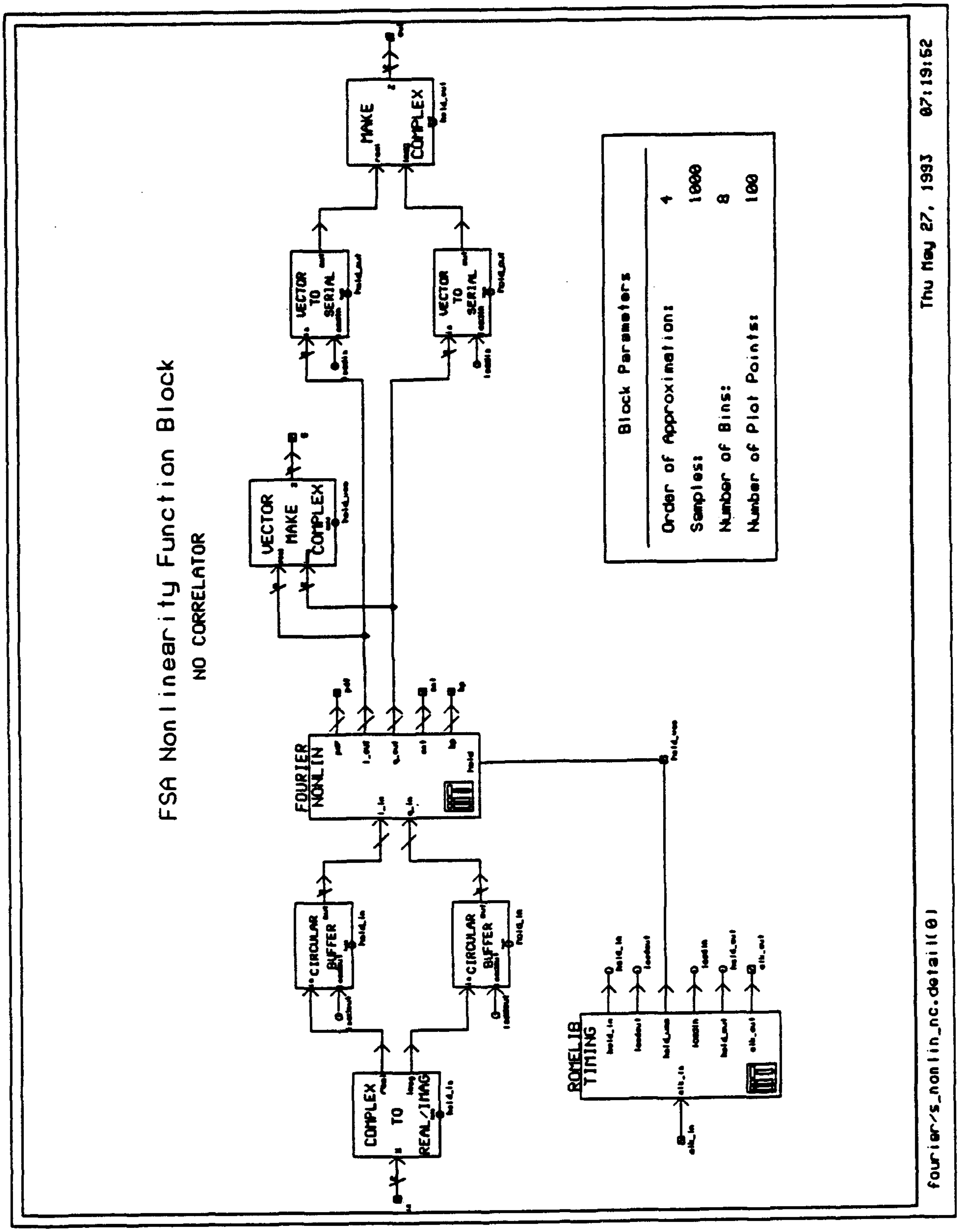


Figure (B-16)

FOURIER/SERIAL_NONLIN

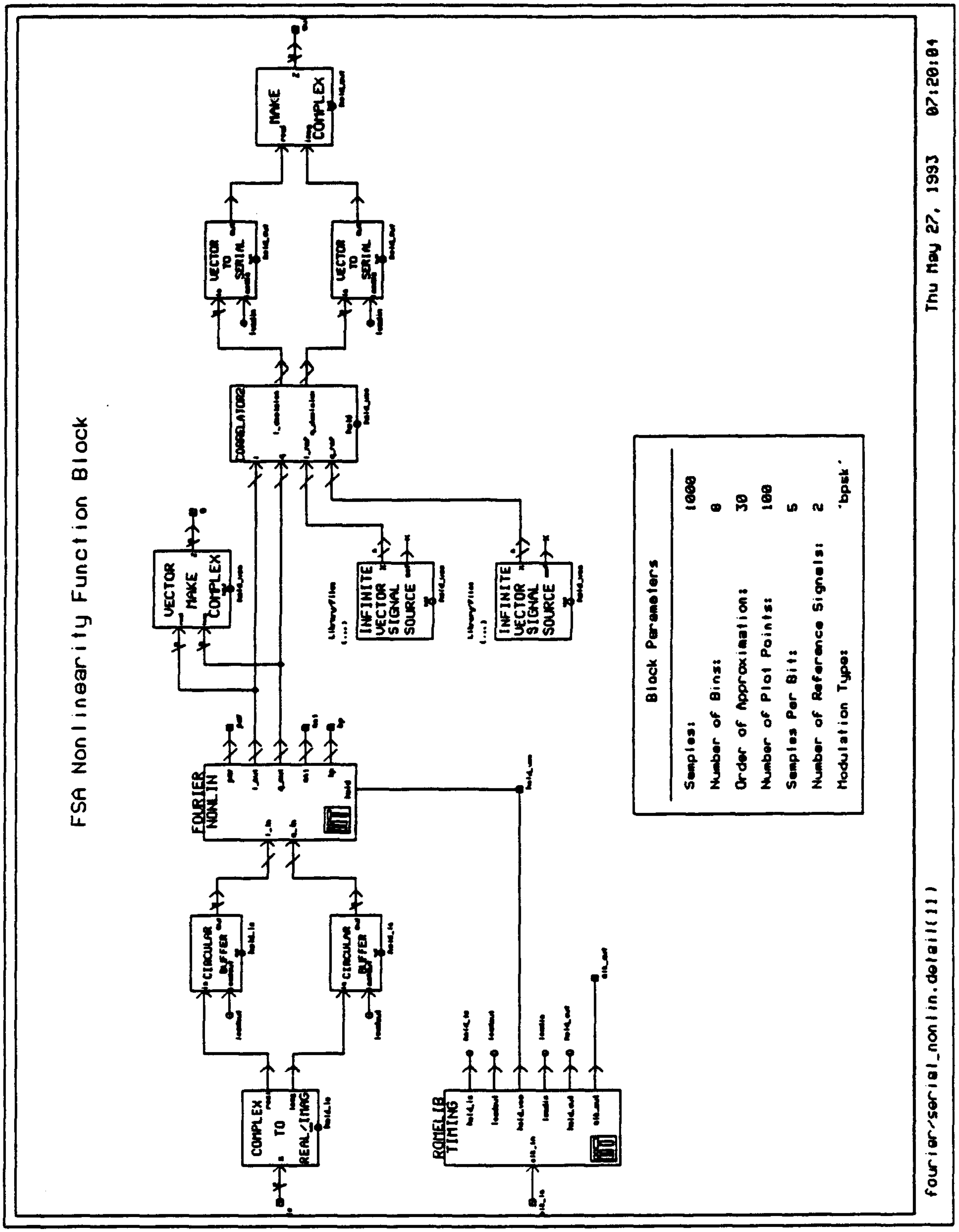


Figure (B-17)

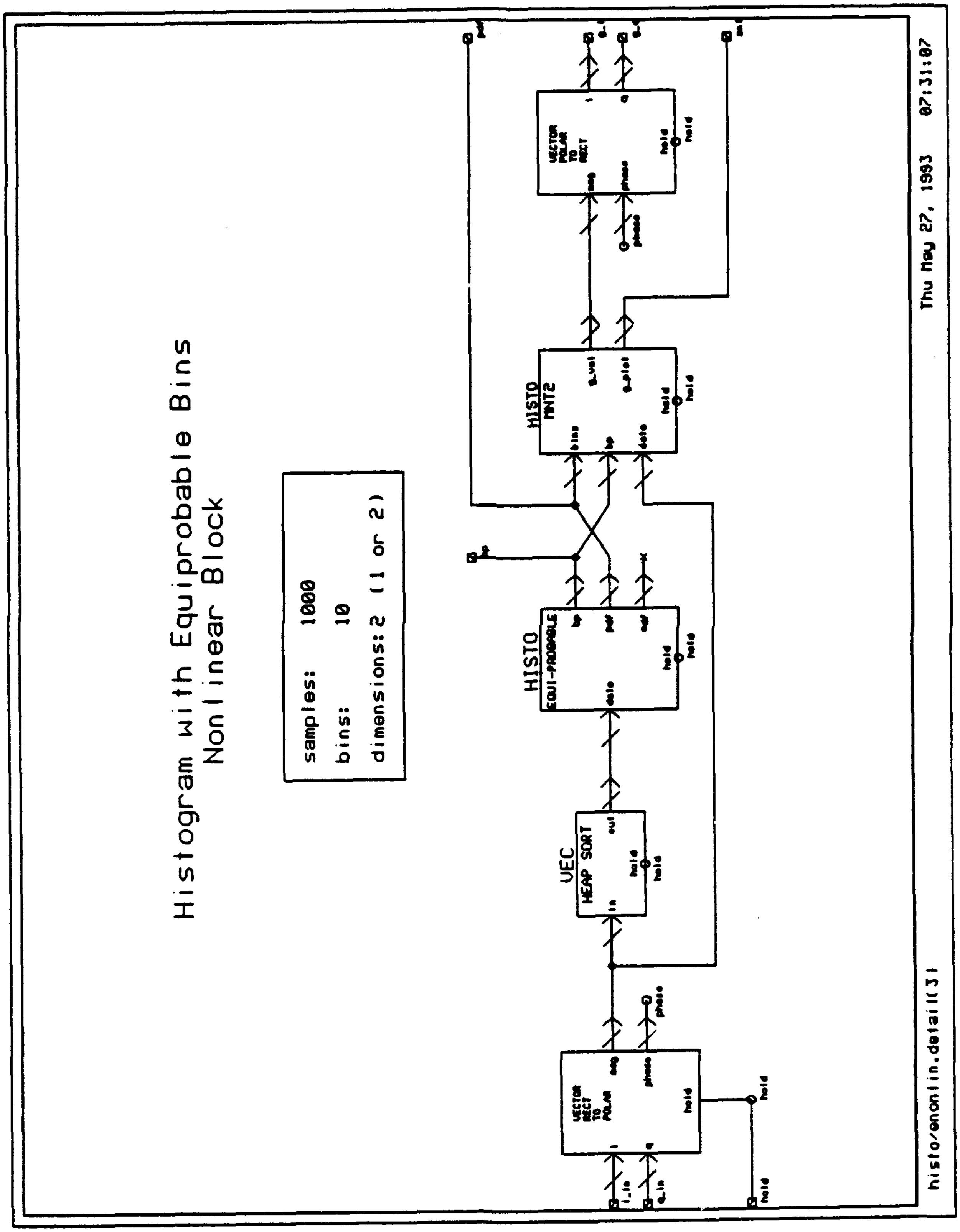

Histogram with Equiprobable Bins
Nonlinear Block
samples: 1000
bins: 10
dimensions: 2 (1 or 2)
VECTOR RECT TO POLAR
VEC HEAP SORT
HISTO EQUI-PROBABLE
HISTO MNT2
VECTOR POLAR TO RECT
Thu May 27, 1993 07:31:07
histo/enonlin.detail(3)

Figure (B-18)

HISTO/MNT2

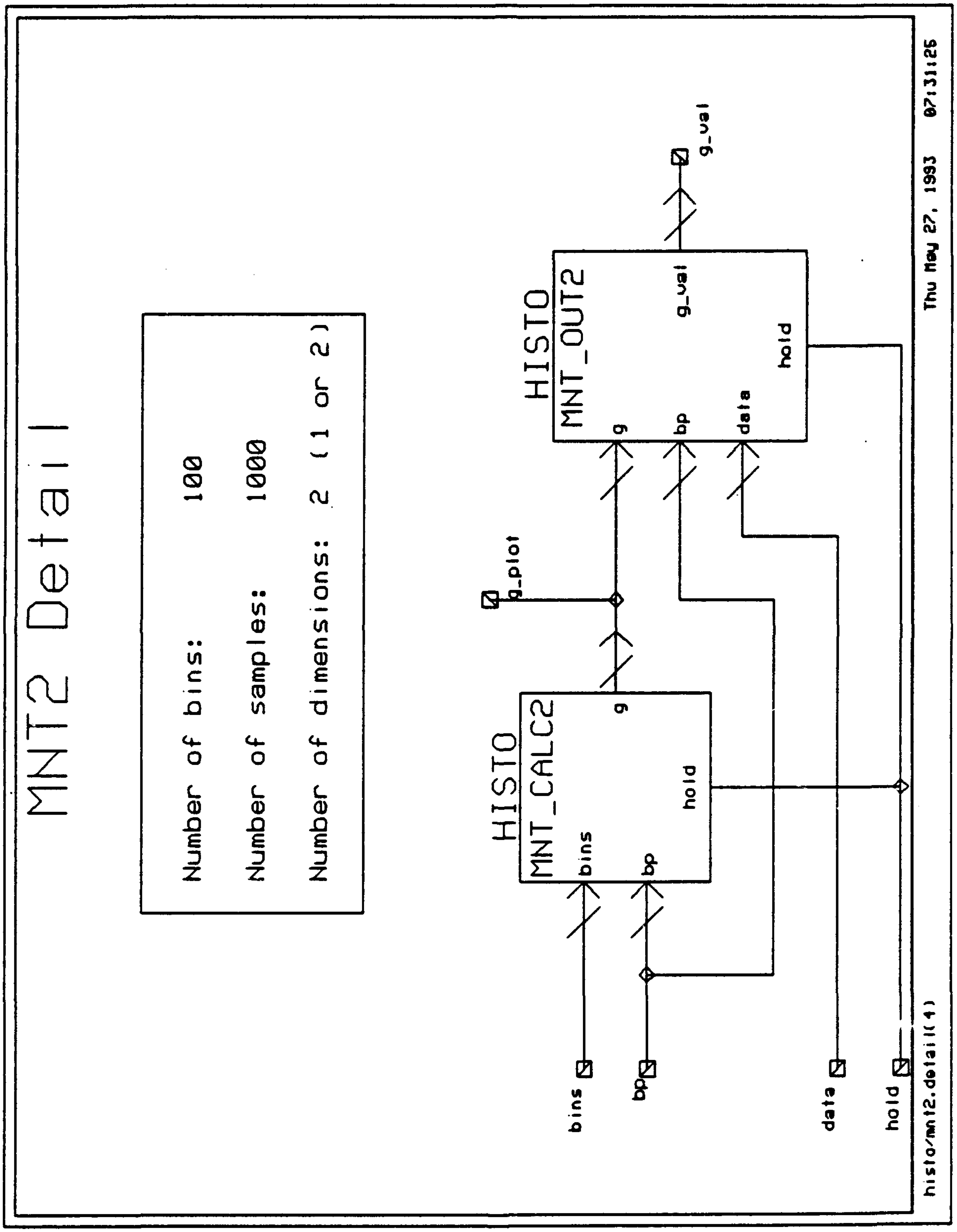


Figure (B-19)

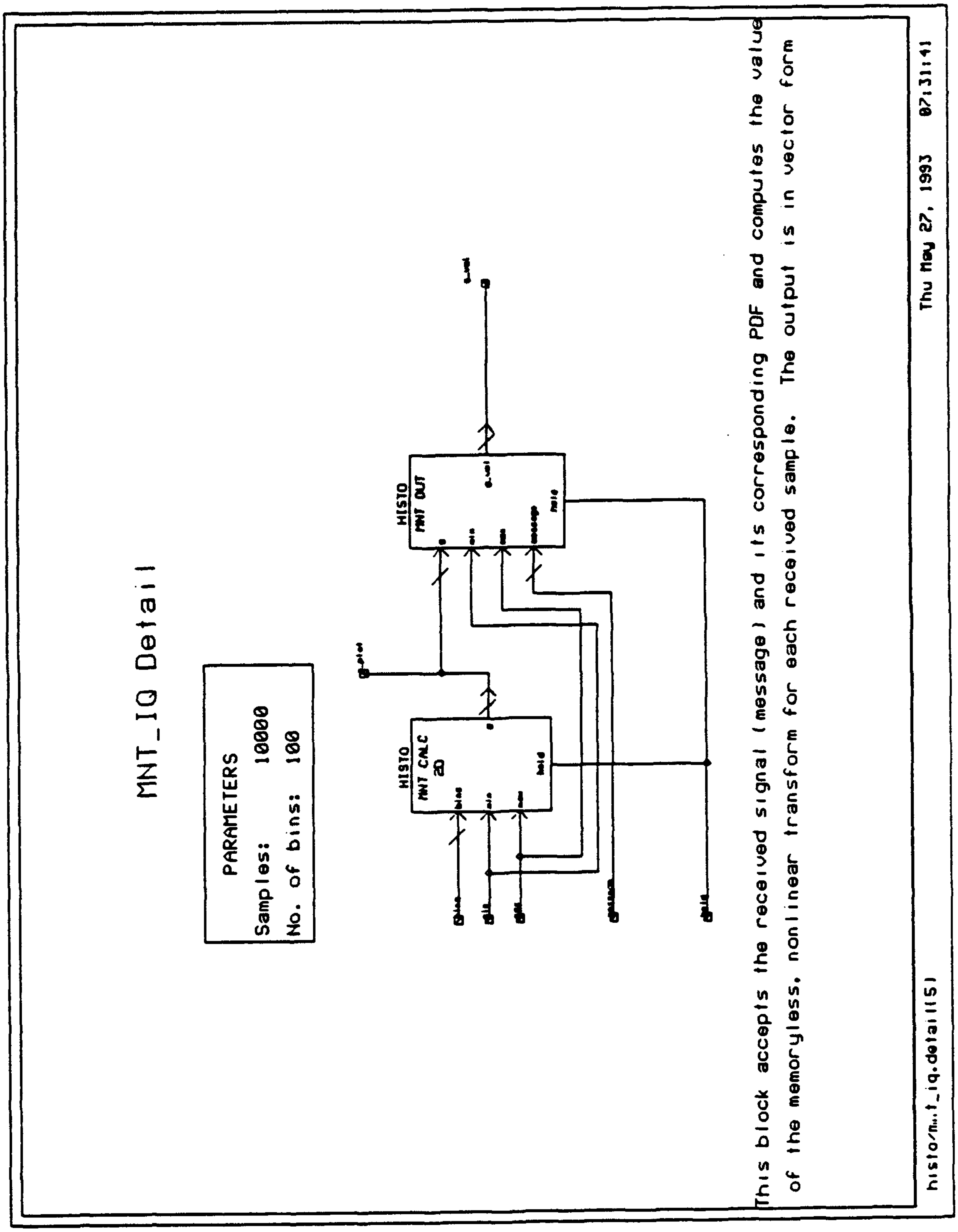

MNT_IQ Detail
PARAMETERS
Samples: 10000
No. of bins: 100
HISTO
MNT CALC
2D
HISTO
MNT OUT
message
hold
This block accepts the received signal (message) and its corresponding PDF and computes the value of the memoryless, nonlinear transform for each received sample. The output is in vector form
histo/m..t_iq.detail(5)
Thu May 27, 1993 07:31:41

Figure (B-20)

HISTO/NONLIN6

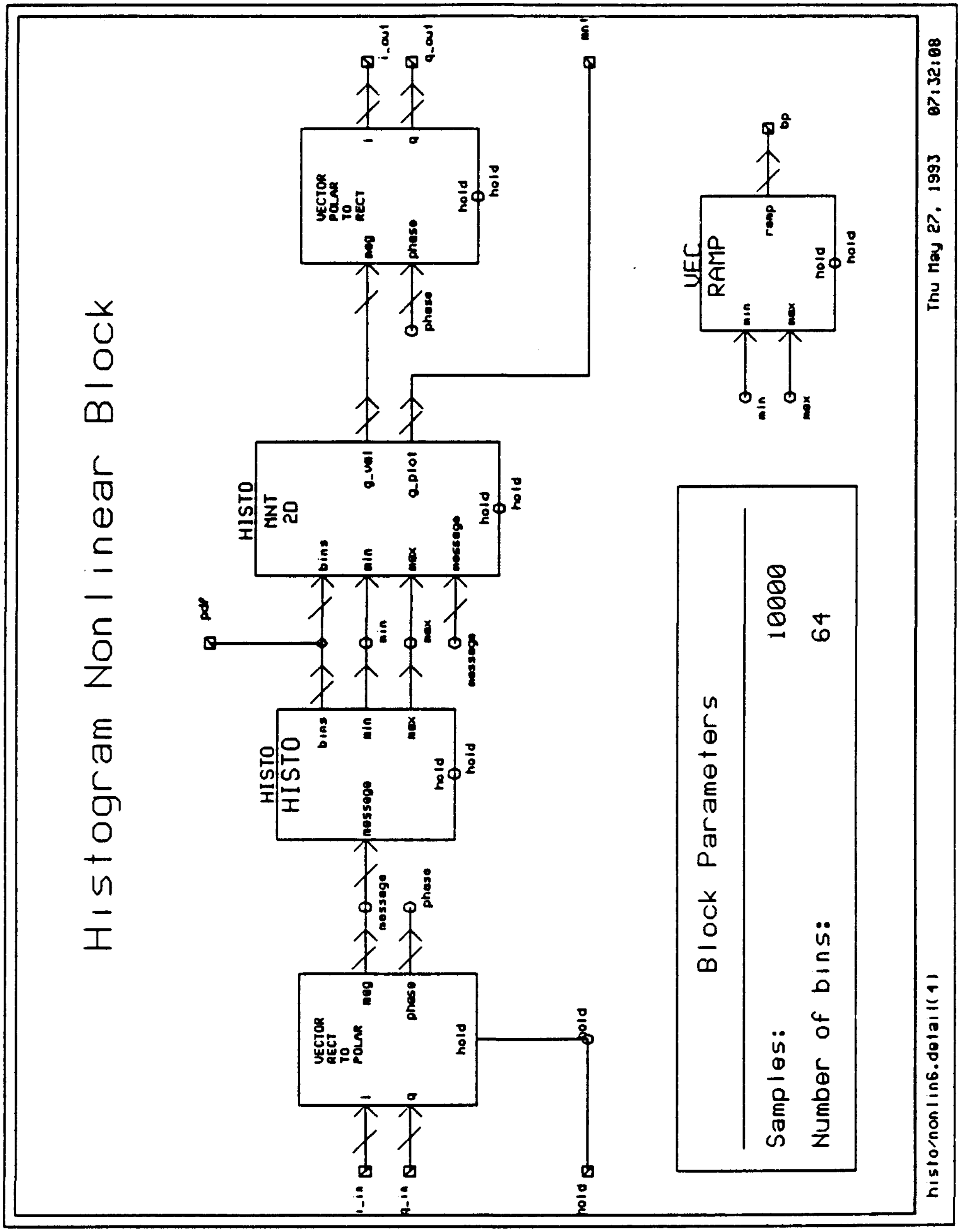


Figure (B-21)

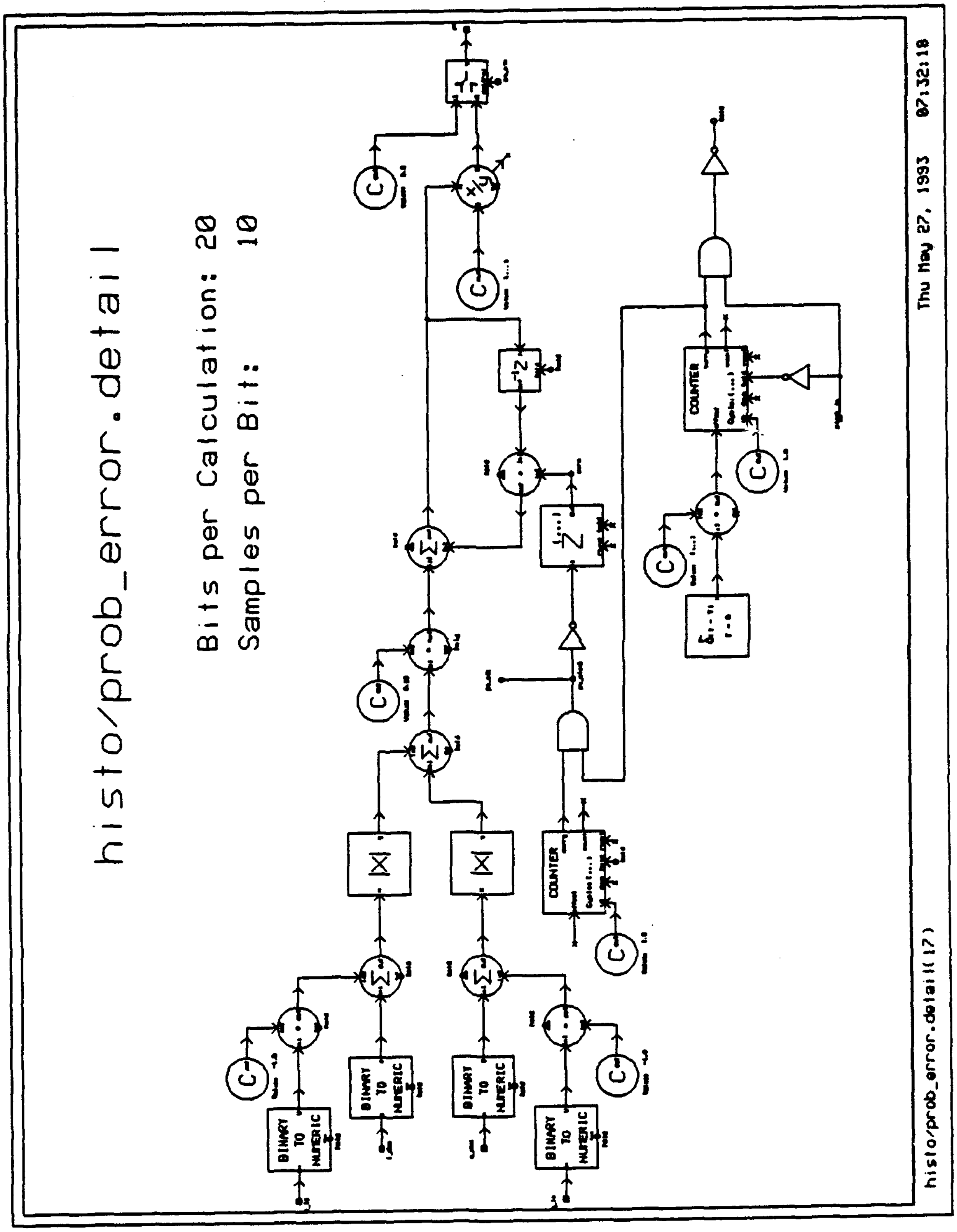
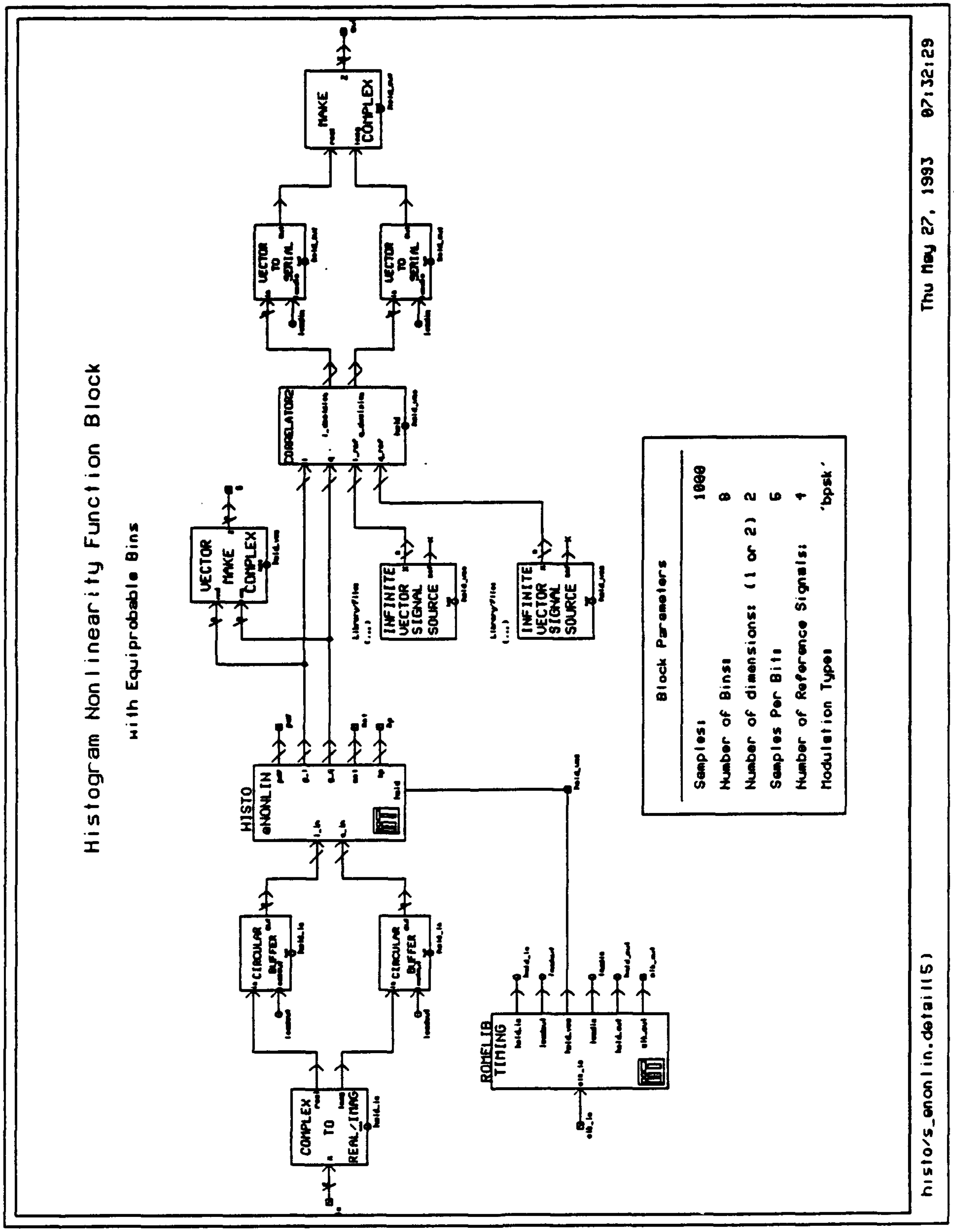

histo/prob_error.detail
Bits per Calculation: 20
Samples per Bit: 10
BINARY TO NUMERIC
BINARY TO NUMERIC
BINARY TO NUMERIC
BINARY TO NUMERIC
COUNTER
COUNTER
Thu May 27, 1993 07:32:18
histo/prob_error.detail(17)

Figure (B-22)



Histogram Nonlinearity Function Block
with Equiprobable Bins
COMPLEX TO REAL_IMAG
CIRCULAR BUFFER
HISTO eNONLIN
VECTOR MAKE COMPLEX
INFINITE VECTOR SIGNAL SOURCE
CORRELATOR2
VECTOR TO SERIAL
MAKE COMPLEX
ROMELIB TIMING
Block Parameters
Samples: 1000
Number of Bins: 8
Number of dimensions: (1 or 2) 2
Samples Per Bit: 6
Number of Reference Signals: 4
Modulation Type: 'bpsk'
histo/s_enonlin.detail(5)
Thu May 27, 1993 07:32:29

Figure (B-23)

HISTO/S_ENONLIN_NC

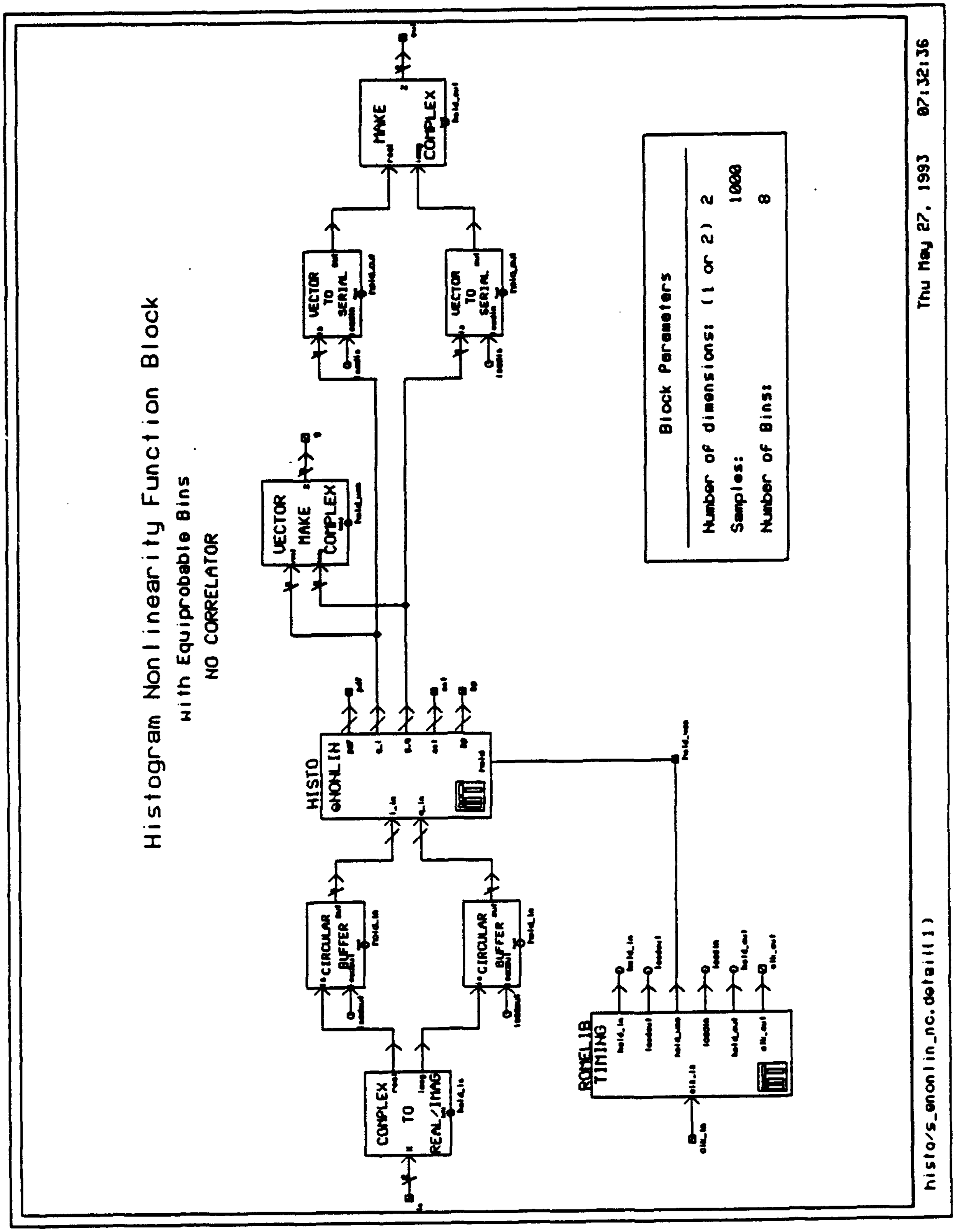


Figure (B-24)

HISTO/S_NONLIN_NC

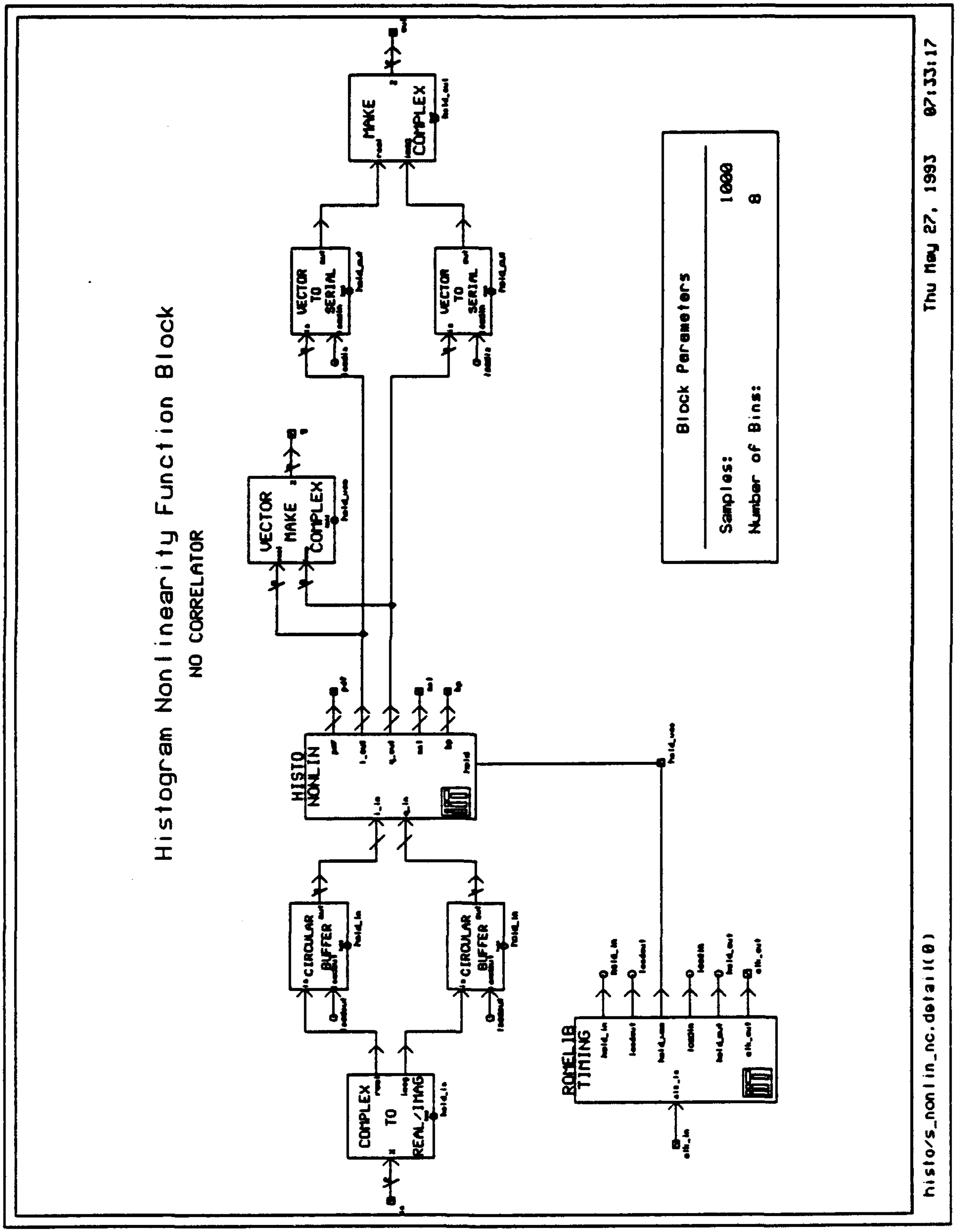


Figure (B-25)

HISTO/SERIAL_NONLIN

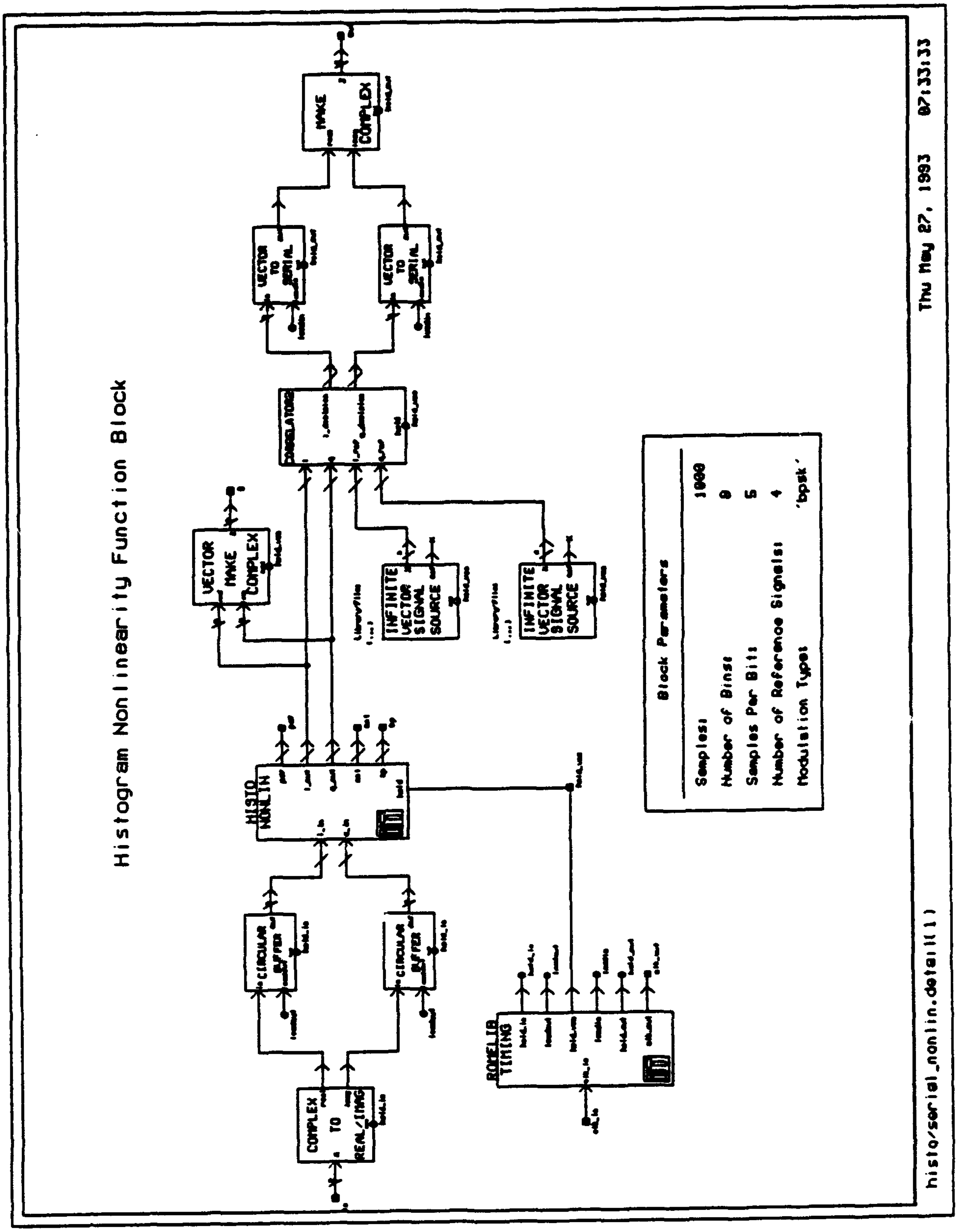


Figure (B-26)

JAM/CHANNEL

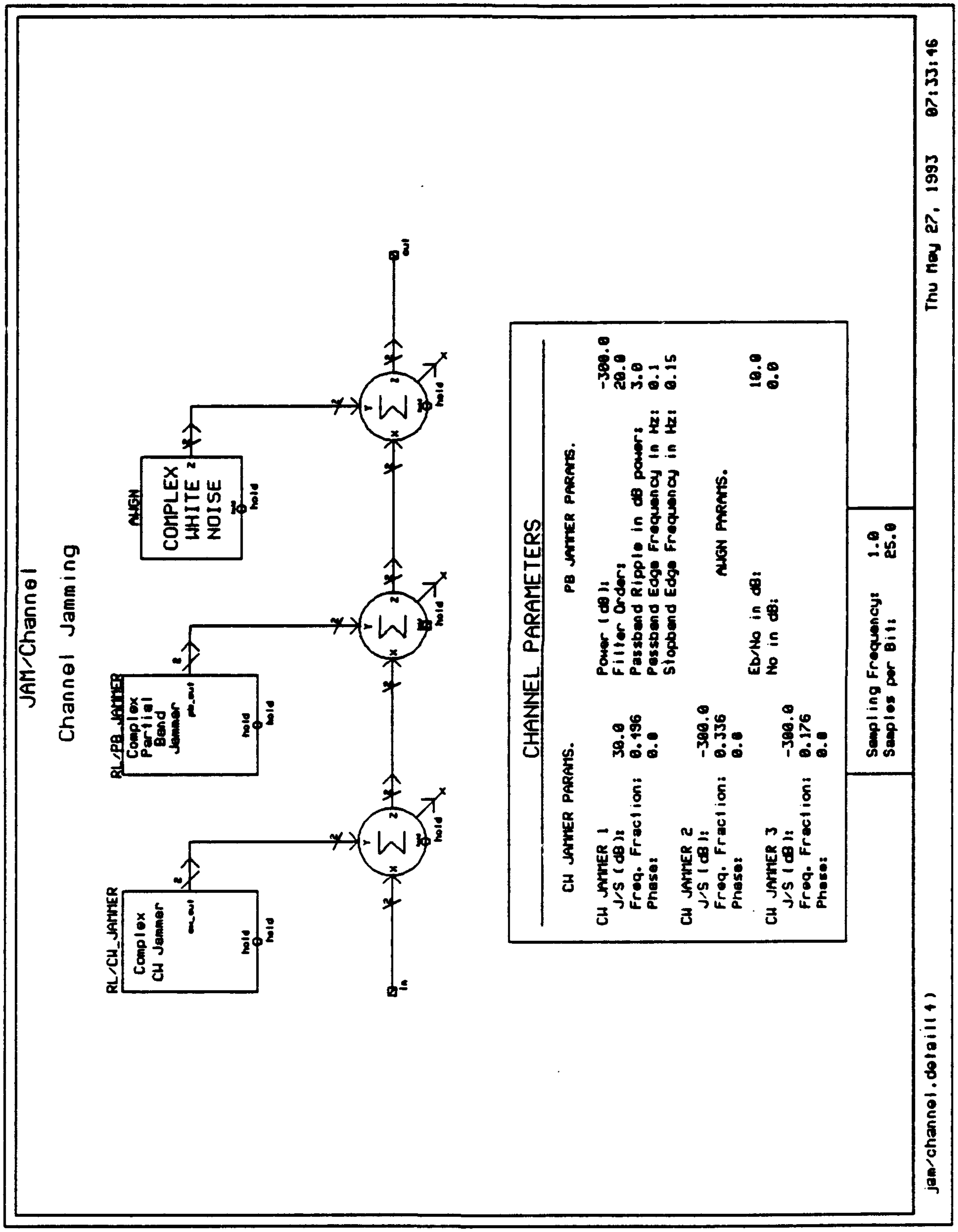


Figure (B-27)

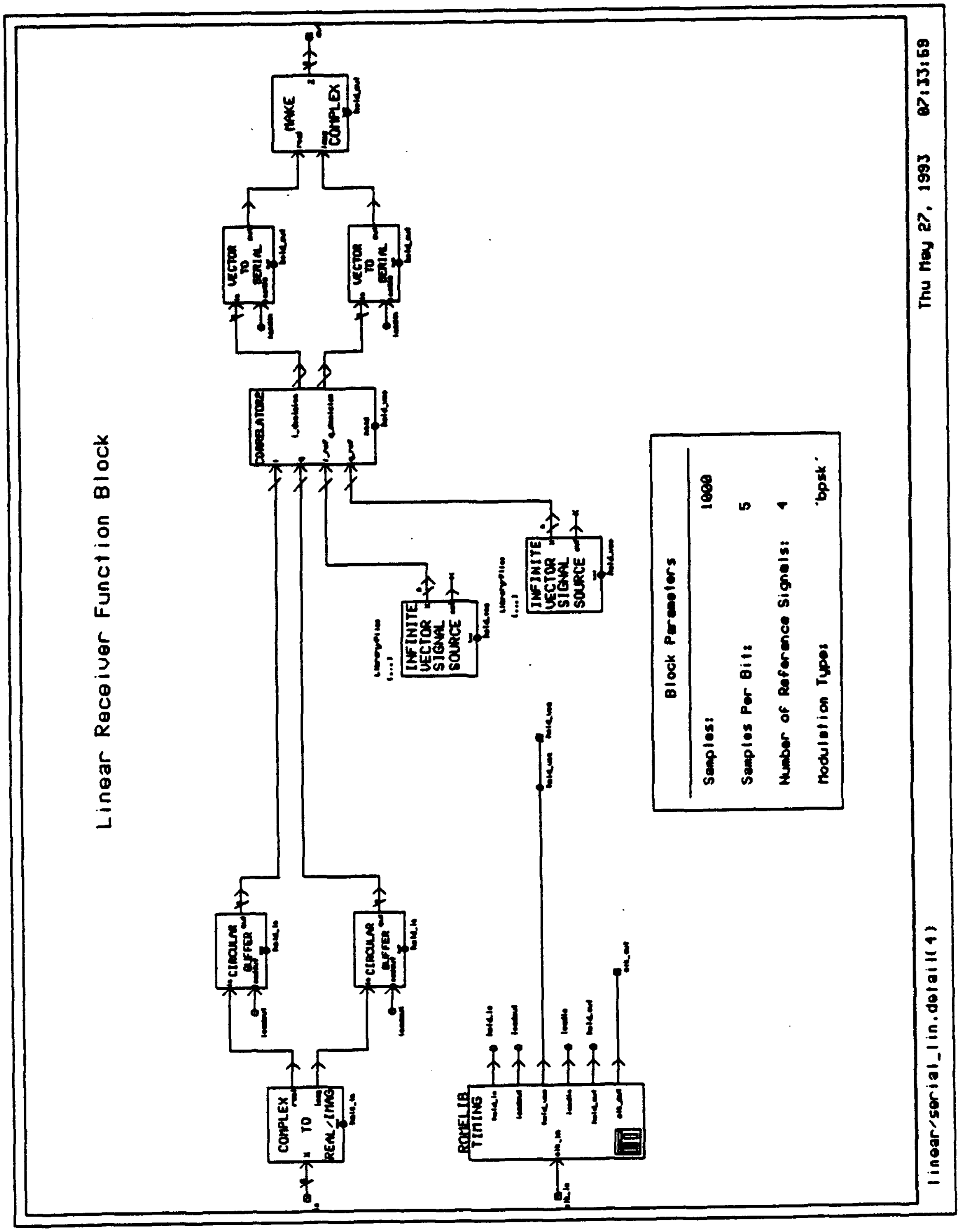

Linear Receiver Function Block
COMPLEX TO REAL_IMAG
CIRCULAR BUFFER
CIRCULAR BUFFER
CORRELATOR2
INFINITE VECTOR SIGNAL SOURCE
INFINITE VECTOR SIGNAL SOURCE
VECTOR TO SERIAL
VECTOR TO SERIAL
MAKE COMPLEX
ROMELIB TIMING
Block Parameters
Samples: 1000
Samples Per Bit: 5
Number of Reference Signals: 4
Modulation Type: 'bpsk'
linear/serial_lin.detail(4)
Thu May 27, 1993 07:33:59

Figure (B-28)

MIPA/NONLIN

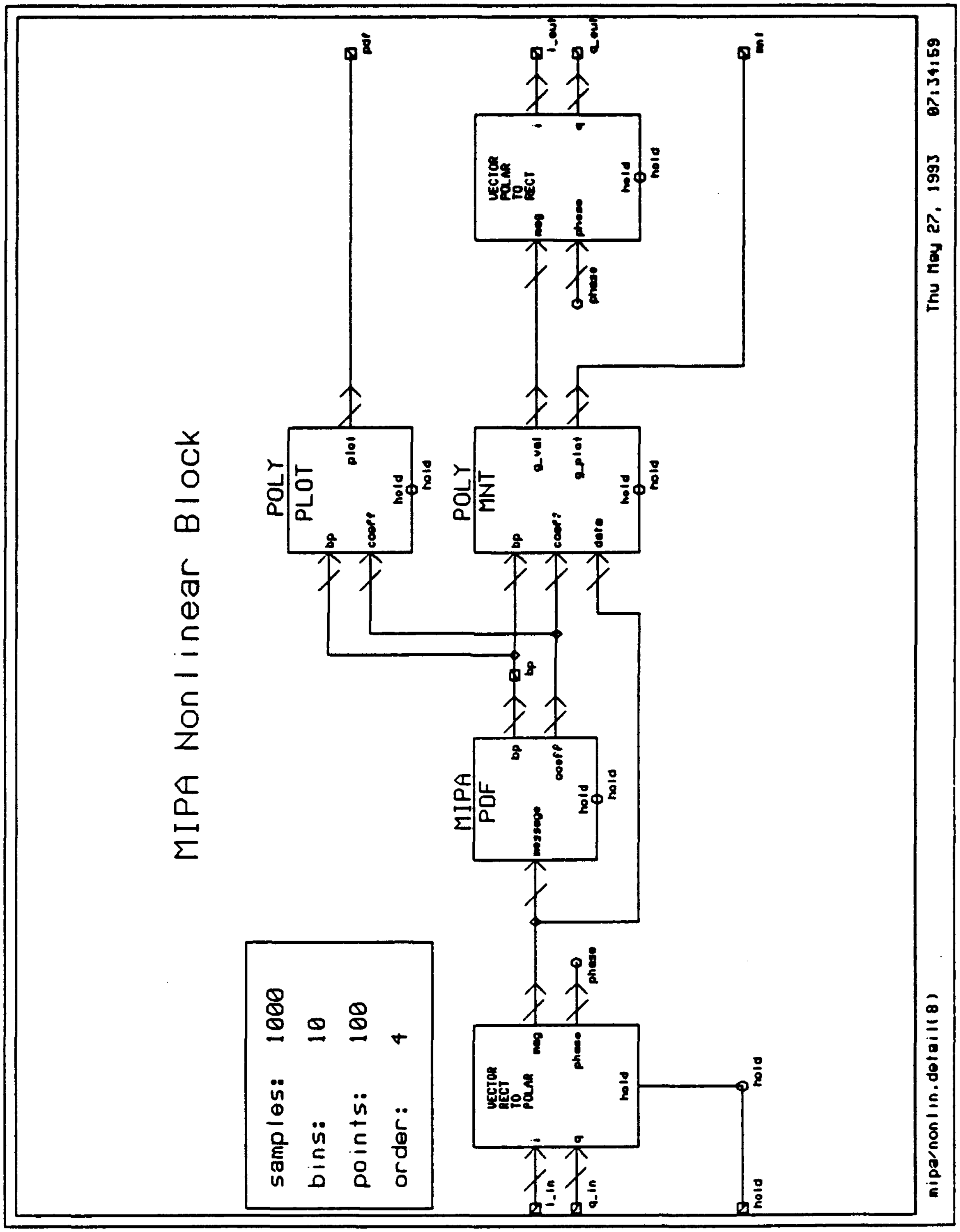


Figure (B-29)

MIPA/S_NONLIN_NC

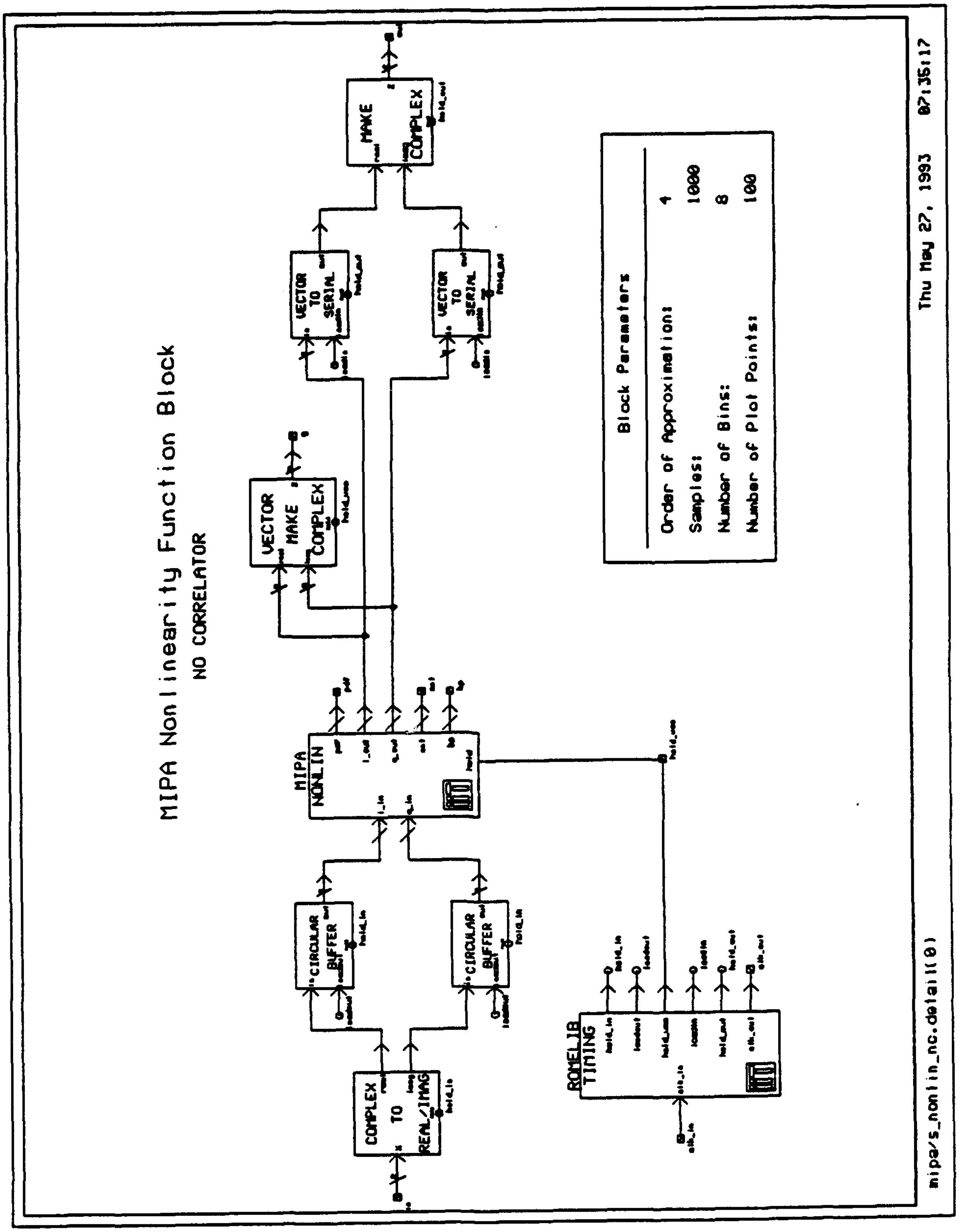


Figure (B-30)

MIPA/SERIAL_NONLIN

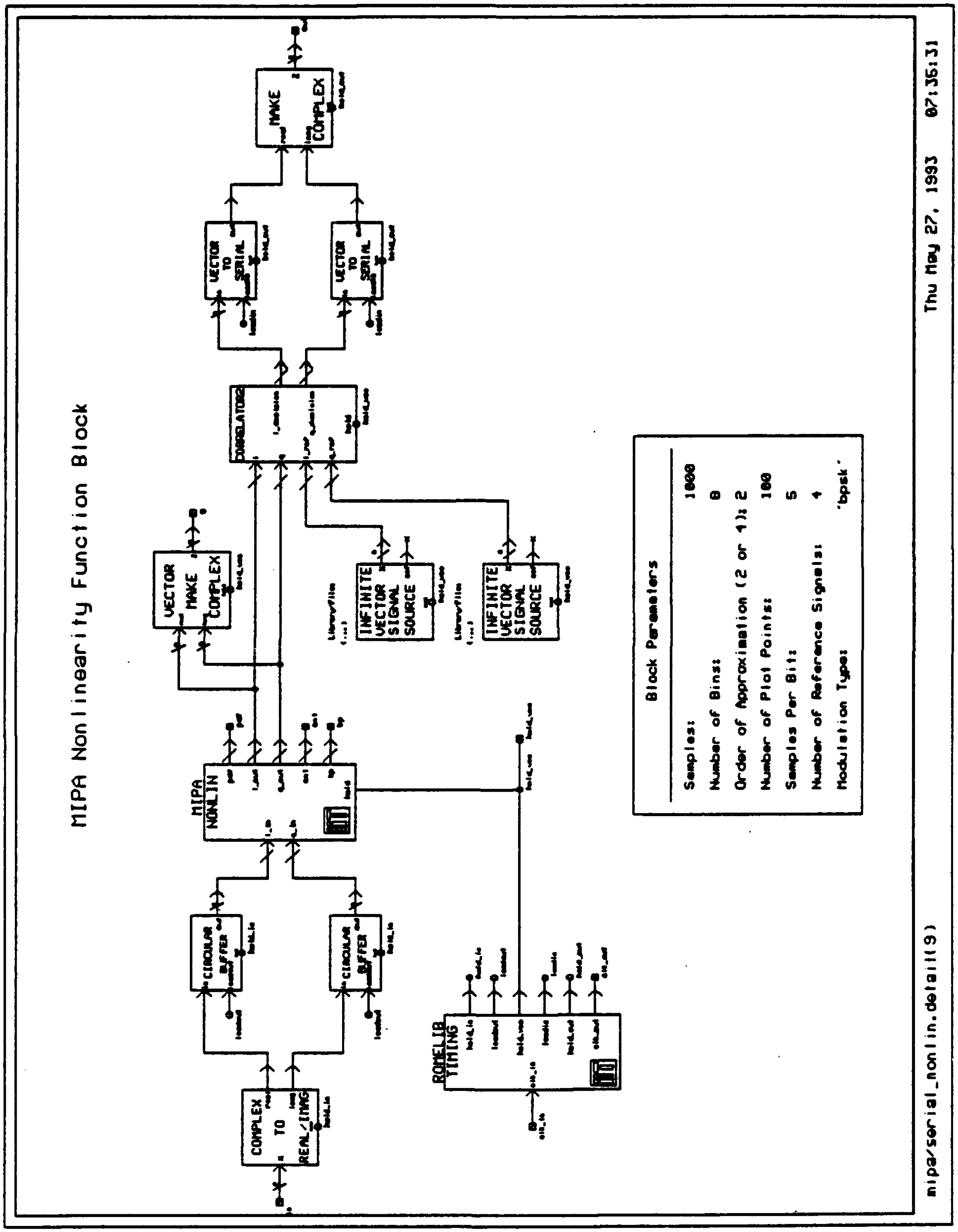


Figure (B-31)

POLY/MNT

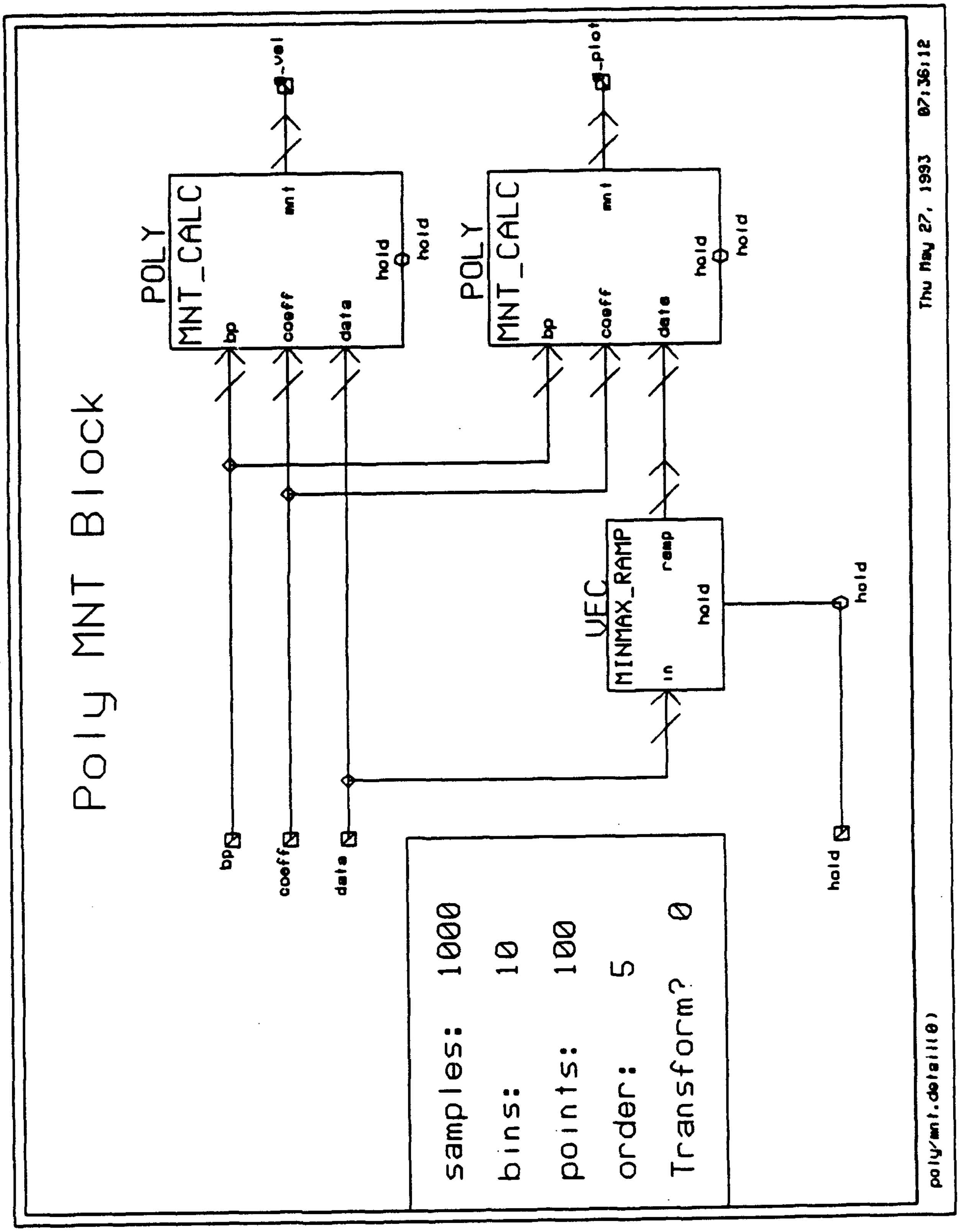


Figure (B-32)

POLY/MNT_GT

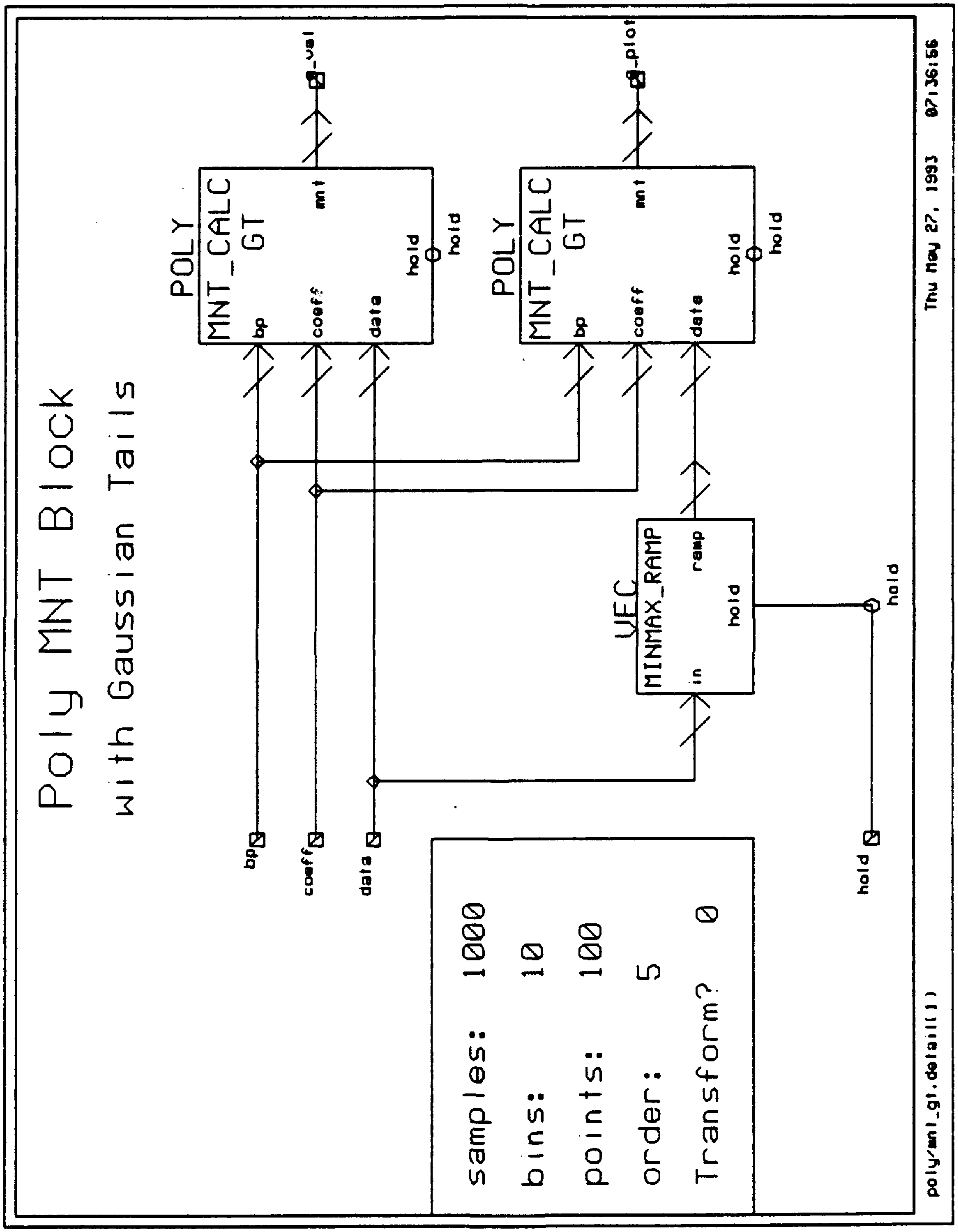


Figure (B-33)

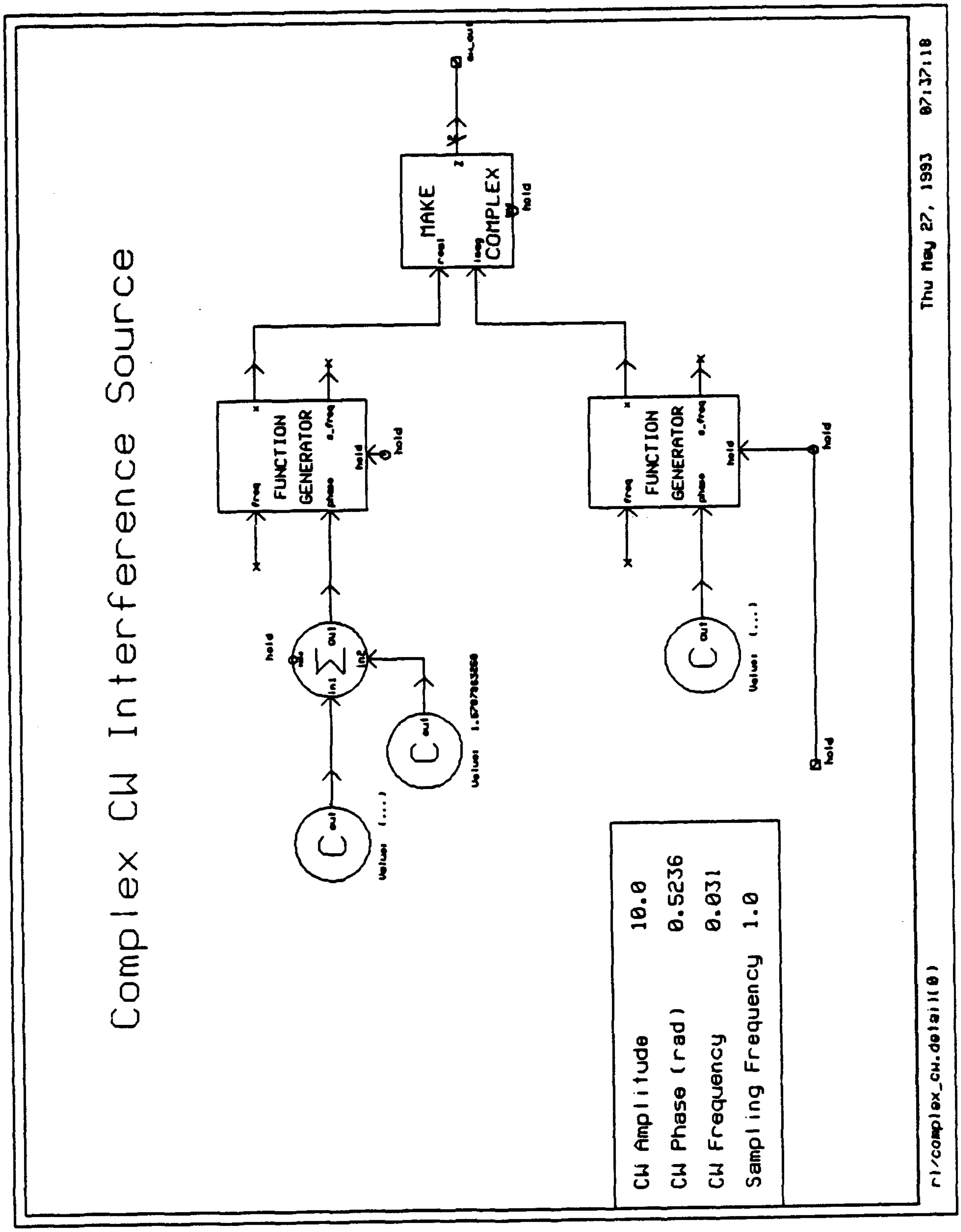

Complex CW Interference Source
FUNCTION GENERATOR
FUNCTION GENERATOR
MAKE COMPLEX
hold
Values 1.5707963268
CW Amplitude 10.0
CW Phase (rad) 0.5236
CW Frequency 0.031
Sampling Frequency 1.0
Thu May 27, 1993 07:37:18
rl/complex_cw.detail(0)

Figure (B-34)

RL/COMPLEX_DATA

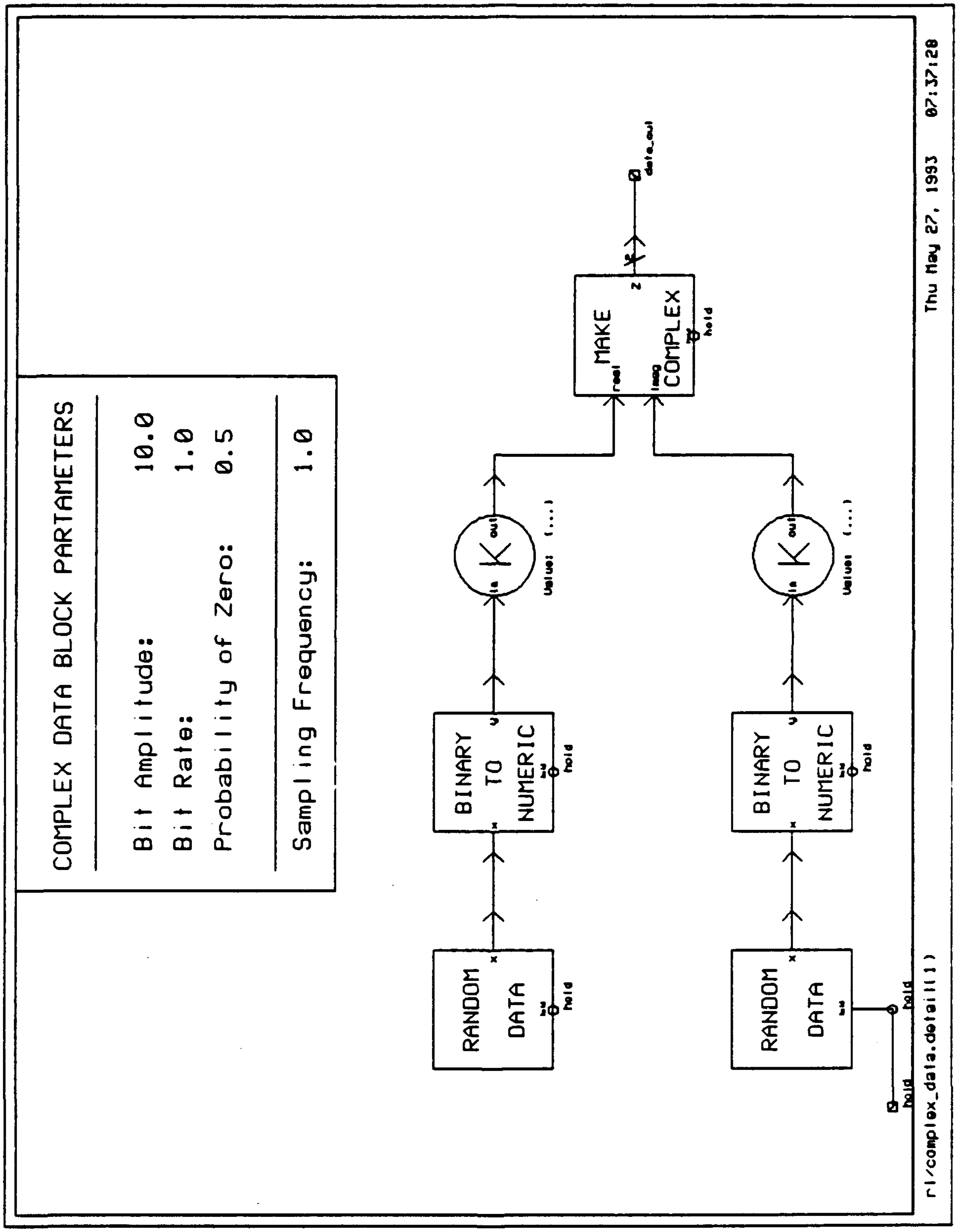


Figure (B-35)

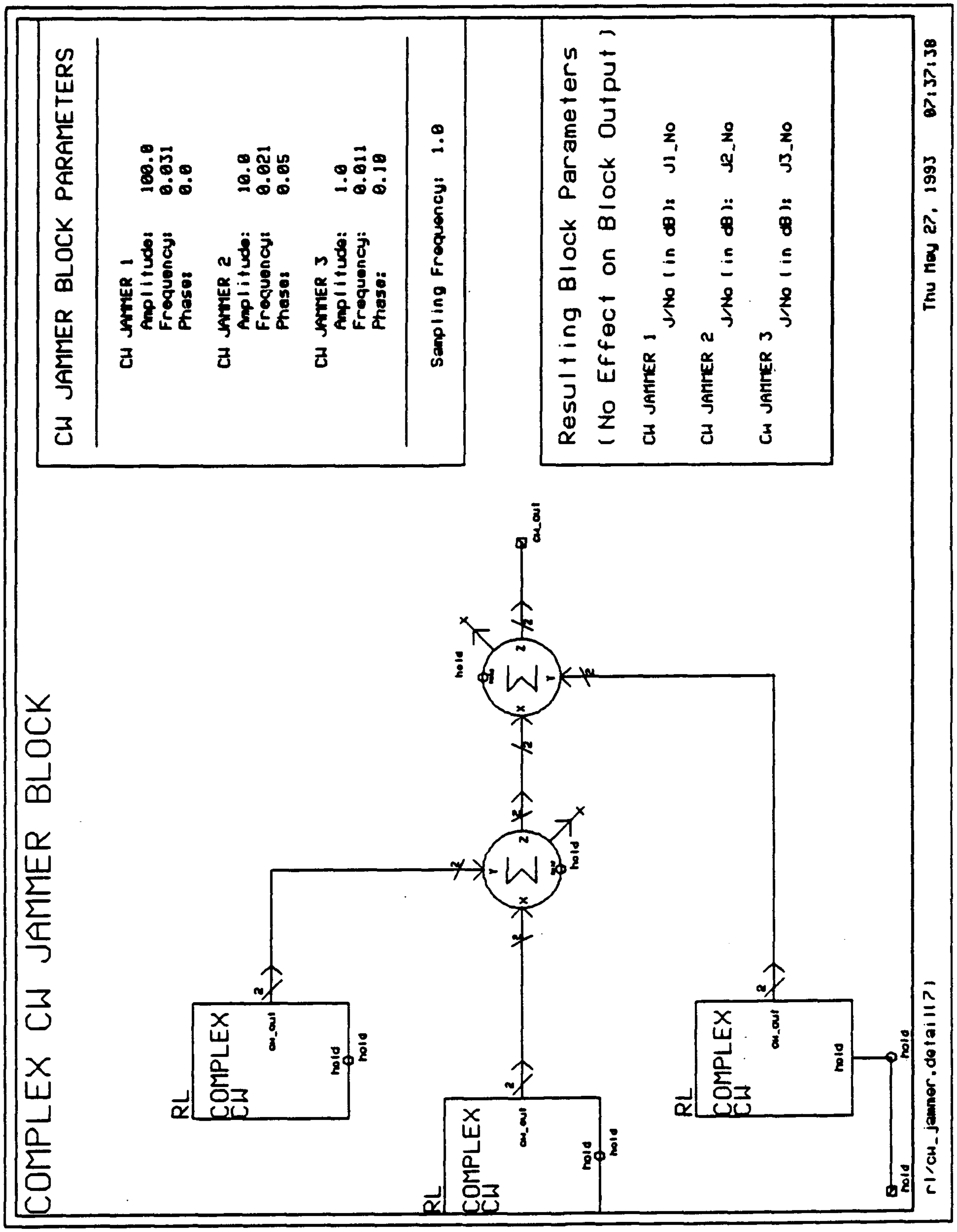

COMPLEX CW JAMMER BLOCK
CW JAMMER BLOCK PARAMETERS
CW JAMMER 1
Amplitude: 100.0
Frequency: 0.031
Phase: 0.0
CW JAMMER 2
Amplitude: 10.0
Frequency: 0.021
Phase: 0.05
CW JAMMER 3
Amplitude: 1.0
Frequency: 0.011
Phase: 0.10
Sampling Frequency: 1.0
Resulting Block Parameters
(No Effect on Block Output)
CW JAMMER 1
J/No (in dB): J1_No
CW JAMMER 2
J/No (in dB): J2_No
CW JAMMER 3
J/No (in dB): J3_No
RL
COMPLEX
CW
cw_out
hold
cw_out
Thu May 27, 1993 07:37:38
rl/cw_jammer.detail(7)

Figure (B-36)

RL/HOLD

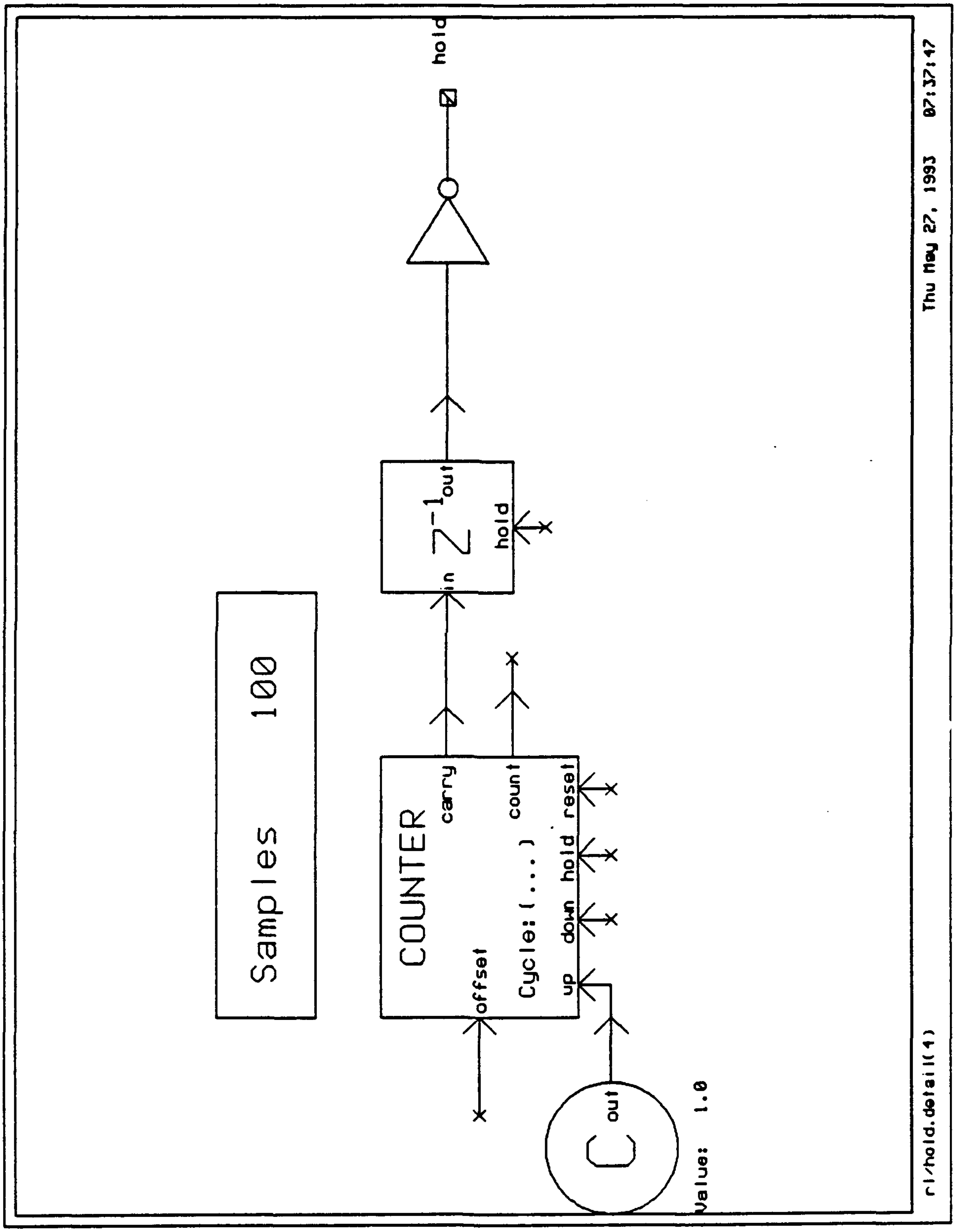


Figure (B-37)

RL/INF_VSOURCE

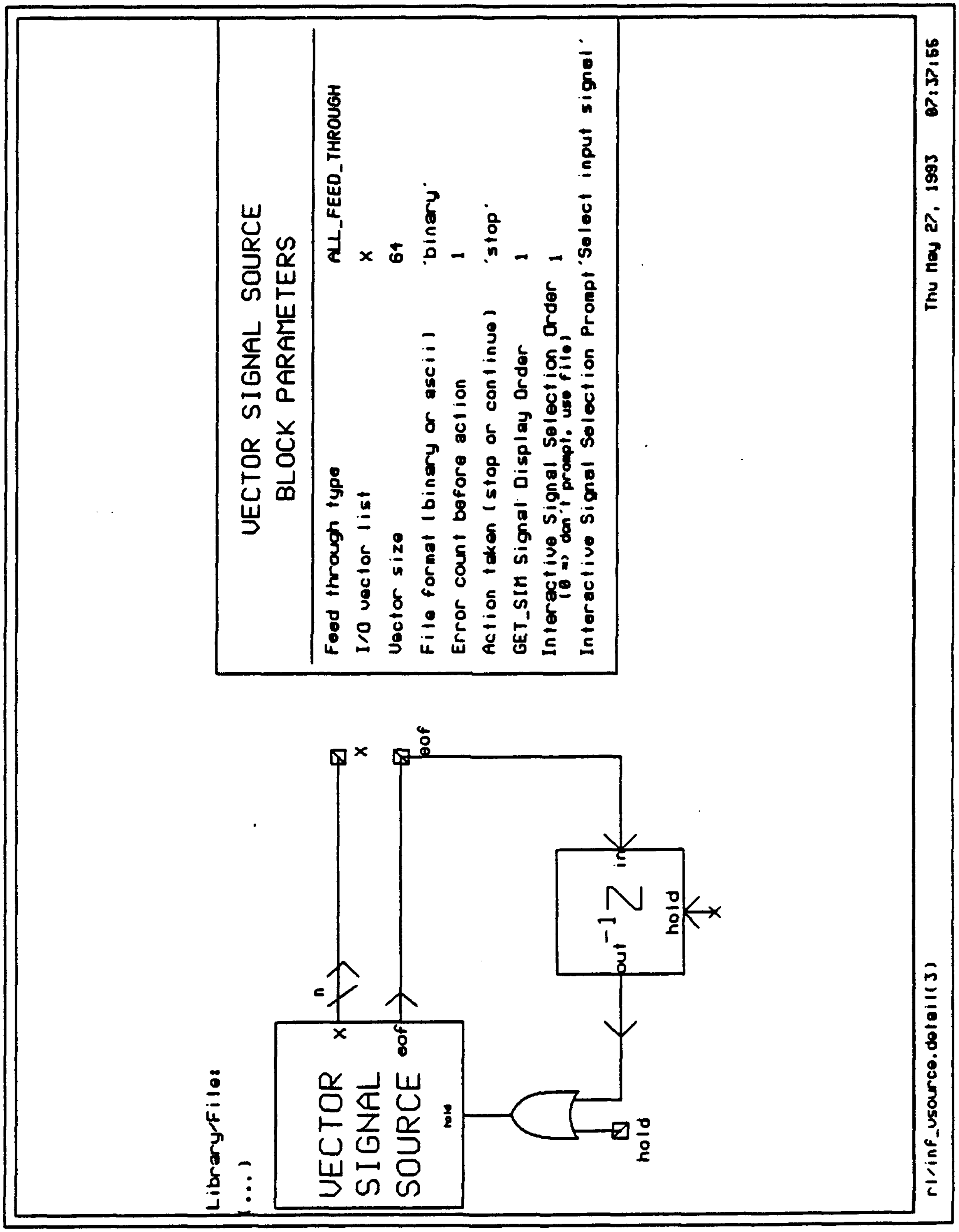


Figure (B-38)

RL/PB_JAMMER

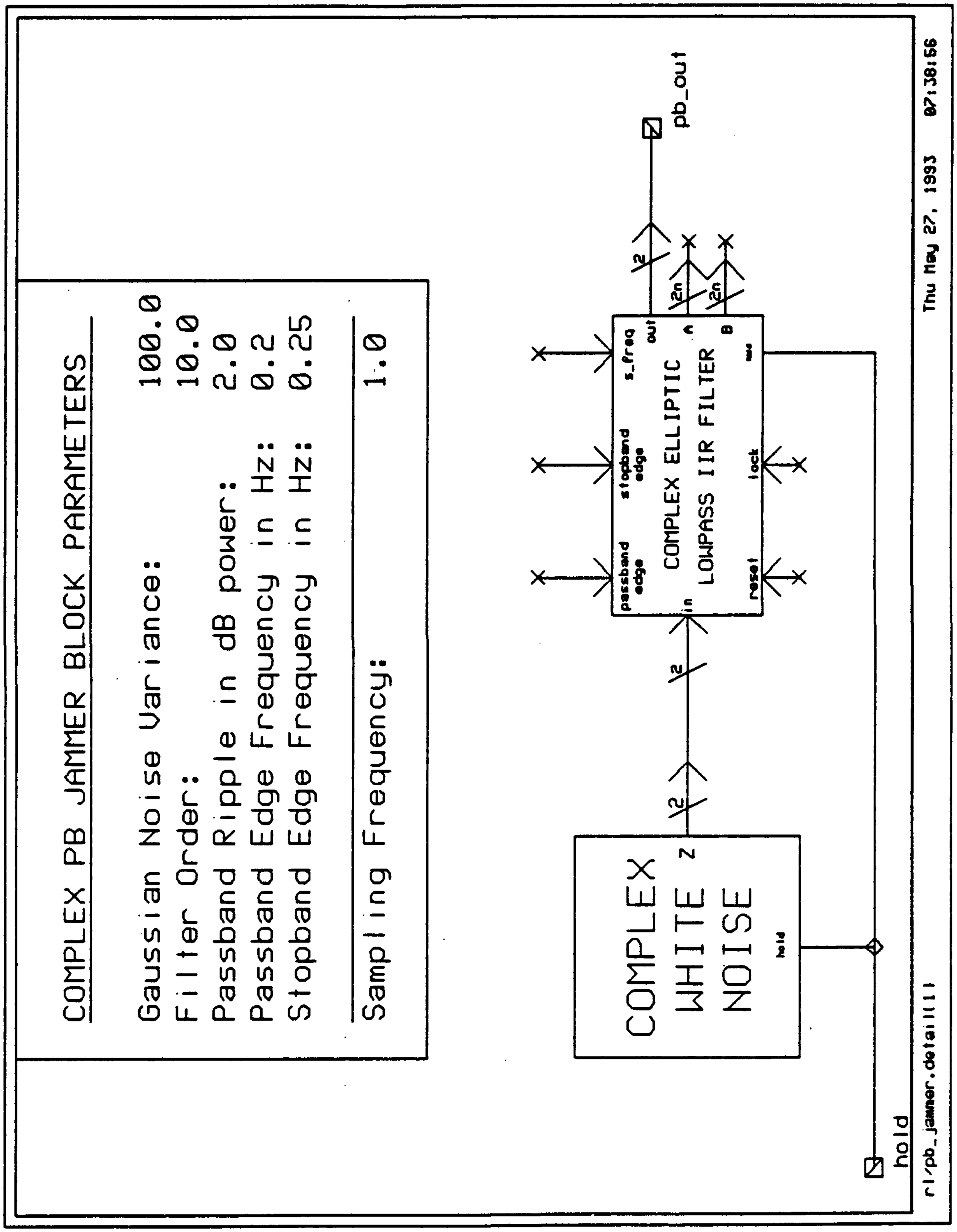


Figure (B-39)

RL/PSK_ERR_CNT

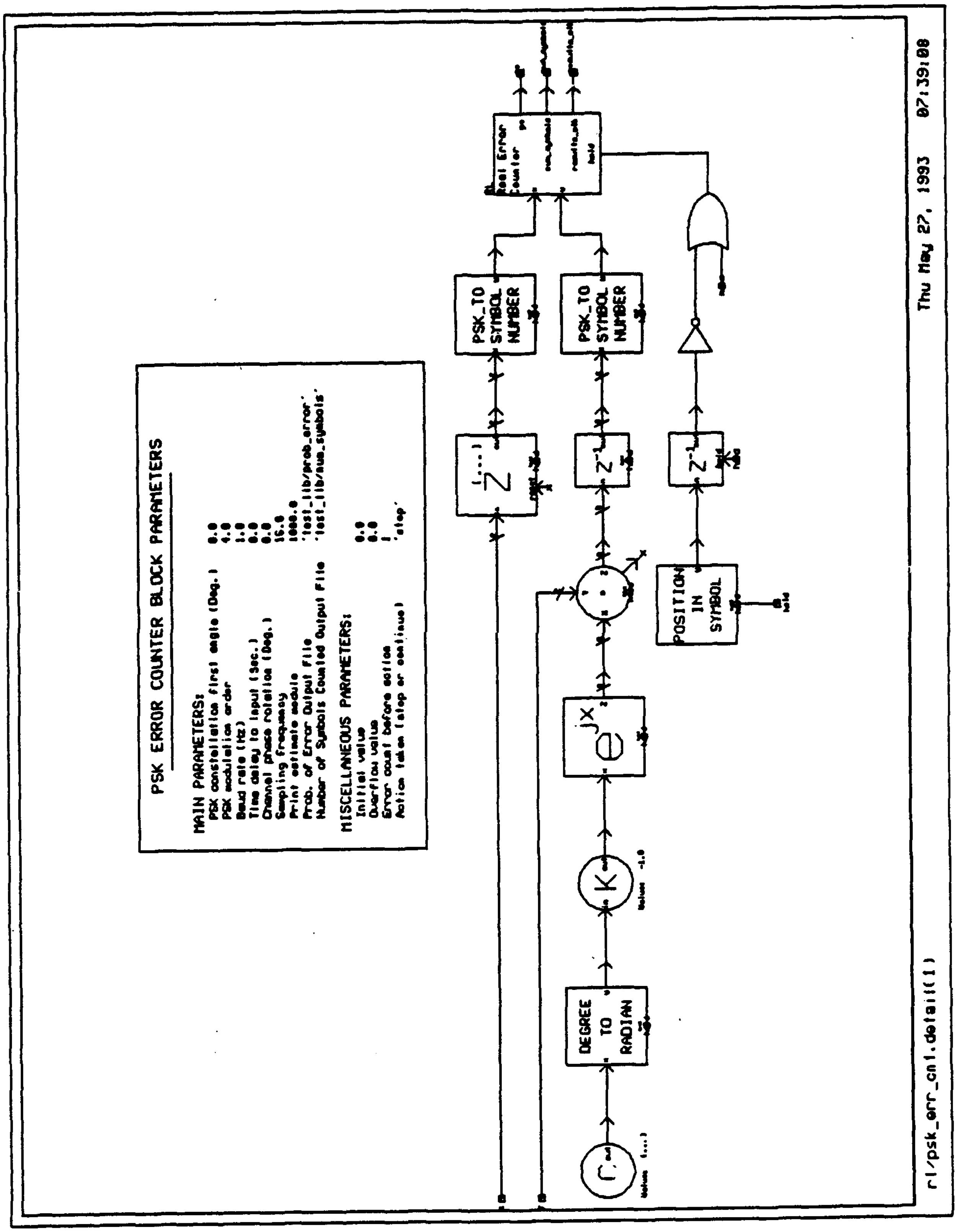


Figure (B-40)

RL/REAL_ERR_CNT

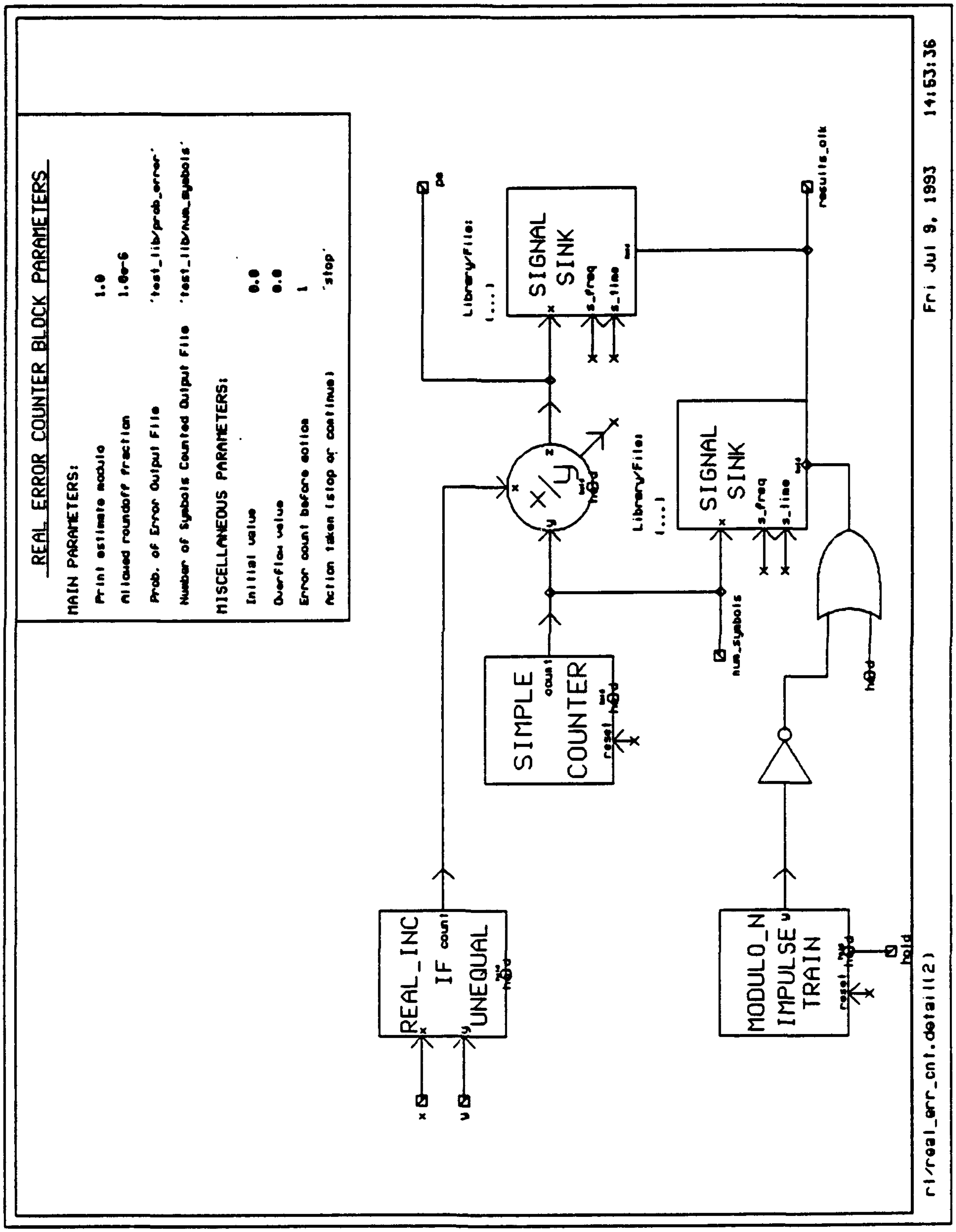


Figure (B-41)

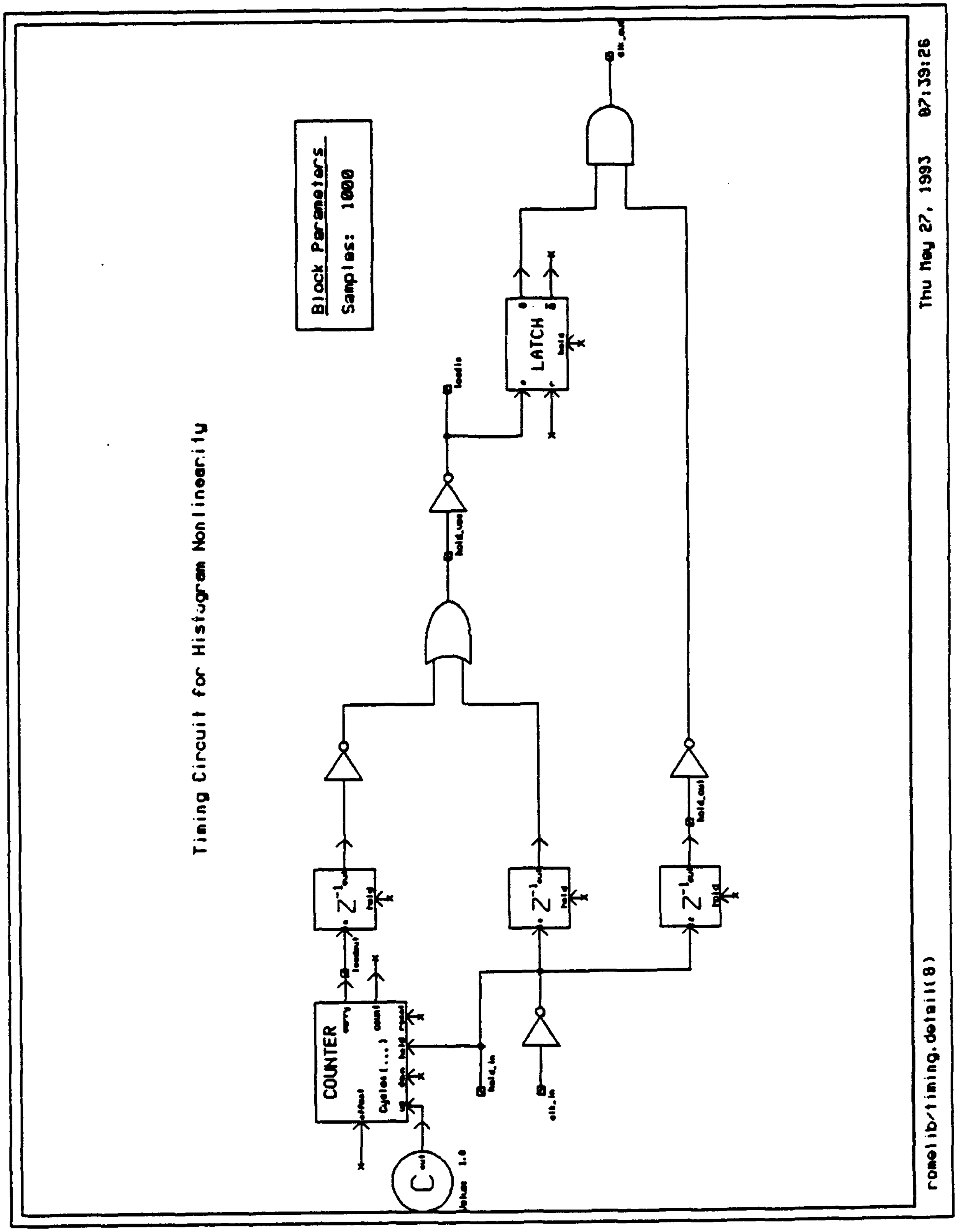

Timing Circuit for Histogram Nonlinearity
Block Parameters
Samples: 1000
COUNTER
LATCH
Thu May 27, 1993 07:39:26
romelib/timing.detail(8)

Figure (B-42)

VEC/MINMAX_RAMP

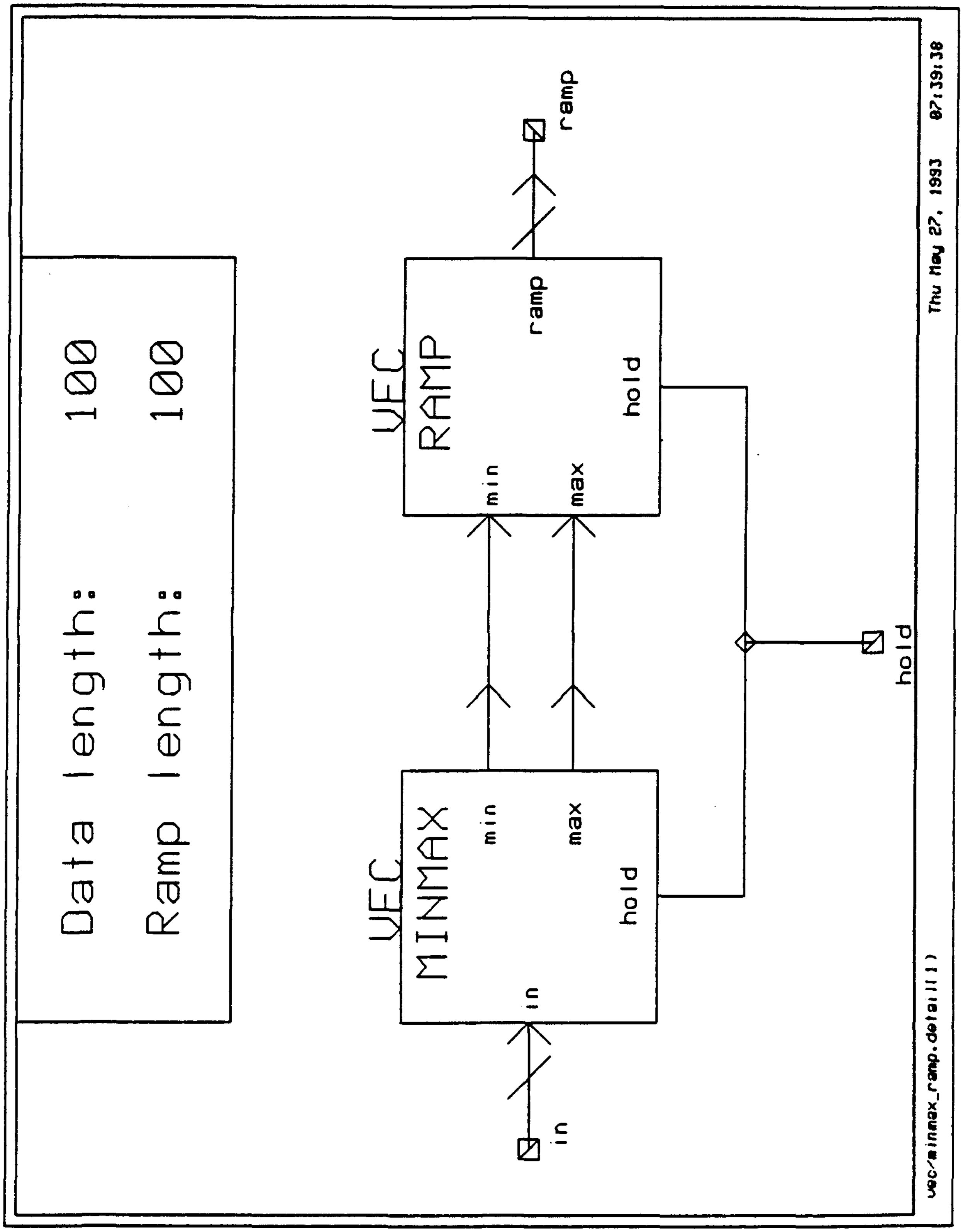


Figure (B-43)

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESC Program Offices (POs) and other ESC elements to perform effective acquisition of C3I systems. In addition, Rome Laboratory's technology supports other AFMC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.